Reconstruction of Gasoline Engine In-Cylinder Pressures Using Recurrent Neural Networks

Colin Bennett

A Thesis Submitted for the Degree of Doctor of Philosophy

University of Sussex
School of Engineering and Informatics
Department of Engineering and Design

June 2013

SUMMARY

Knowledge of the pressure inside the combustion chamber of a gasoline engine would provide very useful information regarding the quality and consistency of combustion and allow significant improvements in its control, leading to improved efficiency and refinement. While measurement using in-cylinder pressure transducers is common in laboratory tests, their use in production engines is very limited due to cost and durability constraints.

This thesis seeks to exploit the time series prediction capabilities of recurrent neural networks in order to build an inverse model accepting crankshaft kinematics or cylinder block vibrations as inputs for the reconstruction of in-cylinder pressures. Success in this endeavour would provide information to drive a real time combustion control strategy using only sensors already commonly installed on production engines. A reference data set was acquired from a prototype Ford in-line 3 cylinder direct injected, spark ignited gasoline engine of 1.125 litre swept volume. Data acquired concentrated on low speed (1000-2000 rev/min), low load (10-30 Nm brake torque) test conditions. The experimental work undertaken is described in detail, along with the signal processing requirements to treat the data prior to presentation to a neural network.

The primary problem then addressed is the reliable, efficient training of a recurrent neural network to result in an inverse model capable of predicting cylinder pressures from data not seen during the training phase, this unseen data includes examples from speed and load ranges other than those in the training case. The specific recurrent network architecture investigated is the non-linear autoregressive with exogenous inputs (NARX) structure. Teacher forced training is investigated using the reference engine data set before a state of the art recurrent training method (Robust Adaptive Gradient Descent – RAGD) is implemented and the influence of the various parameters surrounding input vectors, network structure and training algorithm are investigated. Optimum parameters for data, structure and training algorithm are identified.

Declaration

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for a degree.

Colin Bennett
June 2013

Acknowledgements

This thesis could not have been completed without the help and assistance of a number of people.

I owe particular thanks to my supervisor Dr. J. F. Dunne for his direction, encouragement and enthusiasm throughout my research studies.

My thanks to Mr. Ian Wallis for his knowledge of the engine test facility, and his operation of the test cell for data acquisition. Mr. Barry Jackson's assistance with instrumentation and electrical issues has frequently been invaluable.

I am grateful to Jaguar Land Rover Ltd. for their support both technical and financial for this research programme, and in particular to Mr. David Richardson and Mr. Paul King for their great enthusiasm throughout our many progress reviews.

Finally, I owe much to my wife Beverly, for her encouragement to undertake a PhD, and her great patience and support through the years of study.

Table of Contents

Chapter One

1. INTRODUCTION

The main aim and objective of the work reported within this thesis is the development of an Artificial Neural Network (ANN) structure and training methodology capable of reconstructing in-cylinder pressures within an automotive internal combustion engine using data acquired from sensors external to the engine. The generation of a high quality, reference set of engine operating data to train and test the network is an integral part.

This chapter describes the motivations, details the available measurement methods, and presents the aims and objectives of the work carried out and reported within this thesis. The structure of the thesis is set out and an introduction to the subject area and perceived challenges are given.

1.1 Motivation – Benefits of Cylinder Pressure in Real Time

The internal combustion engine has been the mainstay of the automotive powertrain throughout the industry's history. Despite various diversions into alternative architectures, the reciprocating 4 stroke engine continues to comprehensively dominate the production engine population.

Performance, efficiency and emissions have all improved through consistent development, but the fundamental mechanics are unchanged from the technology's inception with combustion of a hydrocarbon / air mixture providing a contained gas pressure to drive a crank-slider mechanism. Understanding of, and ability to control the combustion event has improved dramatically and contributed significantly to the remarkable improvements both of specific power output, efficiency and emissions reduction achieved over recent years.

A key capability during engine research and development is direct measurement of in-cylinder pressures. Knowledge of this engine parameter

enlightens understanding of ignition point, combustion quality, burn rates and associated energy release and offers major control adaption and optimisation opportunities. Calibration of the engine management strategy is today a major activity in any engine development programme and is significantly assisted by cylinder pressure data measured in real time. Both gasoline (spark ignition) and diesel (compression ignition) engine control strategies could benefit from real time cylinder pressure data.

Multi-cylinder gasoline engines in particular have variable combustion behaviour under part load conditions, with significant volumetric efficiency variations and in-cylinder air motion differences between cylinders, and between cycles during part-throttled operation. The ability to adjust fuelling and ignition timing to balance these variations is attractive, and efficiency could be improved by operating closer to the knock limit than current control methods allow. [Muller et al. 2000], [Yoon et al. 2000].

Diesel engines face very demanding emissions legislation, where engine out nitrogen oxides (NOx) levels in particular are functions of maximum cylinder pressures, and rates of pressure rise. In recent years production engine transducers have been implemented on a small number of engines [SAE Tech Briefs 2007] to assist in achieving the required performance, but are undesirable from cost, complexity and durability aspects.

Control of advanced intake boosting architectures and alternative combustion regimes, such as Homogenous Charge Compression Ignition (HCCI) or Controlled Auto Ignition (CAI), would also benefit from cylinder pressure information. There is also potential to manage the combustion event for improved Noise, Vibration and Harshness (NVH), particularly if higher frequency components of the pressure trace can be identified. Many of the diagnostic features required for On Board Diagnostics (OBD) could also be improved were pressure traces available (e.g. misfire detection), and the development effort for in-vehicle calibration could be reduced with torque estimation available from a cylinder pressure trace [Park, Sunwoo 2003].

An array of in-cylinder pressure instrumentation is available for laboratory use, but few are appropriate for installation in production engines owing to high cost, limited robustness and low durability. Despite these challenges, such are the benefits of in-cylinder pressure data that several production diesel vehicles are now using pressure sensors (Audi A8, VW Jetta, Vauxhall Zafira) [SAE Tech Briefs, 2007] and VW propose their use in high efficiency gasoline engines [Automotive Engineer Nov. 2012] to assist in achieving the 95g/km CO_2 emission fleet average targets proposed for European car sales in 2020.

The cost and durability issues are challenging, and as a result, the ability to reconstruct cylinder pressure using data acquired from sensors external to the engine is highly attractive, and previous work has sought to achieve this using engine cylinder block structural vibration [Vulli 2006], [Bizon et al. 2011], crank kinematics [Potenza 2005] [Hamedovi et al. 2005] and spark ignition ionisation current [Lee et al. 2001] among others. In particular, crankshaft rotational position sensors and cylinder block vibration transducers (knock sensors) are frequently already fitted and have proven durability with acceptable cost. All of the above measures however have non-linear transfer functions to cylinder pressure with responses dependant on both amplitude and frequency contents, resulting in functions that vary with both engine speed and load hence making an inverse model challenging to construct.

Crankshaft angular acceleration is proportional to the resultant applied torque, but even if the only torque component present were sourced from the cylinder pressure, directly inverting the crank acceleration to cylinder pressure is not straight forward. The crank kinematic equations are set out fully in section 2.3, but the simplified version of equation (2.9) relating cylinder pressure to its component of torque is given below in equation (1.1), and it is immediately clear that re-arranging the equation to evaluate P_g results in $sin\theta$ appearing in the denominator, and the equation becomes singular at TDC when θ and $sin\theta$ are zero. The primary point of interest in the pressure signal – P_{max} – typically occurs close to TDC.

$$T_g \approx P_g \pi \frac{b^2}{4} \mathrm{r} \sin\theta \left(1 + \frac{r}{l} \cos\theta\right) \quad (1.1)$$

where:
T_g = torque component on crankshaft due to cylinder pressure
P_g = cylinder pressure acting on piston crown
b = cylinder bore diameter
r = crank throw (half stroke)
l = connecting rod length
θ = crankshaft angle from top dead centre

Artificial Neural Networks (ANNs) by contrast can provide a powerful and efficient approach to nonlinear function estimation, and previous work has looked to leverage their capabilities towards the reconstruction of cylinder pressure [Potenza et al, 2007]. This work seeks to improve on the capability of an ANN for cylinder pressure reconstruction, to develop more effective network structures and to apply a recently developed training algorithm for a recurrent architecture, exploring the efficiency and reliability of the training process.

The work completed for this thesis has generated a reference set of operating data from a running engine including synchronous in-cylinder pressures, crank kinematics and cylinder block vibration. A recent development in training of recurrent ANNs has been implemented and applied to this engine dataset to reconstruct cylinder pressure on unseen data with acceptable results.

1.2 Methods of Cylinder Pressure Measurement and Reconstruction

Here several direct methods of cylinder pressure measurement are discussed, focusing on the particular installation arrangement including flush mounted sensors and those integrated with the spark or glow plug. Indirect methodologies for pressure reconstruction are then described.

1.2.1 Direct Measurement

The majority of laboratory measurement of in-cylinder pressure for research and development purposes is conducted using direct measuring piezo-electric pressure transducers. Laboratory grade transducers are expensive,

in the order £1000-£2000 per unit (at 2012 prices), and require a charge amplifier to convert the transducer output to a measureable voltage signal – with a typical cost of £1000 per channel. Both transducer and amplifiers require regular calibration, and great care is necessary with the cabling between the transducer and amplifier as the very low level charge signal is prone to noise interference. The installation of these transducers can take a number of forms:

'Flush mounted'

The transducer is installed such that the sensing element is flush with the cylinder head gas face. Usually this requires a specially machined bore to be cut into the head to carry the transducer. On most engines this will need to pass through the coolant jacket, making positioning and sealing difficult and costly to achieve. With the face of the transducer flush with the gas face, there are no transfer chambers or mechanical resonances to corrupt the signal – such an installation is regarded as the best available, and is important where higher frequency components of the pressure signals are important such as for combustion noise studies and development of NVH reduction strategies.

Integrated into a spark plug

A special spark plug, with an integrated pressure transducer replaces the standard spark plug; as a result installation is straight forward. The position of the electrode gap may be compromised compared to the standard spark plug, and this type of transducer usually has a limited frequency response compared to the flush mounted type – it is however entirely adequate for the frequency range important for crank kinematic studies.

Diesel glow-plug replacement

For development purposes, this transducer installation is held by an adapter which replaces the cold start glow plug in a diesel engine. This offers flush mounting of the sensing element and easy installation (though the adapter sleeves are costly), but precludes studying cold start cylinder pressures where the glow plug would normally be active. A number of production diesel engines have also used glow plug based transducers (e.g. the Beru

transducer, installed in Audi A8, Volkswagen Jetta, Vauxhall Zafira and Insignia). Here a pressure derived force is transmitted via a rigid core to an external piezo-electric element with the glow plug function retained.

The inclusion of expensive transducers on production engines is a mark of how challenging current emissions legislation has become, particularly the NO_x requirements of Euro VI and US Bin5 legislation. Diesel engines are better able to carry the additional cost as they demand a significant sales premium. However, the cost remains highly undesirable and in-service durability is as yet unproven.

Optical pressure measurement devices have also been reported, using fibre optic transmission of light, to measure the deflection of a diaphragm in contact with the cylinder pressure. While these may ultimately promise better calibration stability and signal noise behaviour, they still present the general design difficulties of a direct measurement sensor and have not seen any wide use even in the laboratory.

1.2.2 Indirect Measurements for Reconstruction

Several parameters external to the combustion chamber are highly influenced by the combustion event and the associated pressure pulse within. These include:

Crankshaft angular acceleration

Gas pressure acting on the piston crown drives the crank-slider mechanism and causes crankshaft rotational acceleration. The crank acceleration is a function not only of the gas pressure, but also of crank rotational position, engine speed, loading from reciprocating and rotating inertia torques, and friction torques. Crank angular acceleration is relatively easy to measure, but may become unrepresentative of the overall resultant torque input if the operating range includes torsional flexible modes. Crank angular position sensors are generally already available on engines, providing information to control fuel injection and ignition timings. However their typical resolution may not offer sufficiently high resolution of crank kinematics to be useful for the application under consideration in this thesis.

Cylinder block and head vibration

The combustion pressure event results in vibration of the structure of the cylinder block and cylinder head, which can be measured directly using accelerometers suitably positioned on the outside of the block or head. Gasoline engines usually already include a vibration sensor for knock identification, and it is likely this signal could be used to drive reconstruction. However, cylinder pressure is not the only excitation source that causes engine structural vibration, as many other sources of force input are present. In particular, piston slap can be a major contributor to block vibration, and occurs at a similar crank angle as ignition.

Direct fuel injection systems are a further source of vibration excitation. Intake and exhaust valve opening and closing events are also force inputs – these tend to be quite well separated in crank angle from the combustion event for the cylinder under consideration, but on a multi-cylinder engine, valve events from other cylinders may overlap a pressure event. Separating the vibration signatures of the various inputs and extracting the resulting response from the combustion event is one of the challenges for this approach. Additionally, the transfer function from multiple cylinders to a single vibration response point may not be equal, potentially requiring individual reconstruction processes for each cylinder. This measurement is also called structure-borne sound by some researchers [Villarino and Bohme, 2004].

Spark plug ionisation current

The chemical process involved in the combustion event produces charged ions within the cylinder and by applying a bias voltage across the spark plug after the ignition event, an ionisation current can be measured. The combustion pressures and temperatures will influence the ions present through the event, and hence the ionisation current measured can be related to the cylinder pressure. This approach is not considered in detail in this work. Clearly this approach can only be considered for spark ignited engines.

Indirect reconstruction strategies

Various approaches are applied to these signal sources to reconstruct cylinder pressure. Broadly these include:

- Physical models of the engine which are inverted to reconstruct cylinder pressure
- Transfer function approaches identifying a frequency or time domain relationship between cylinder pressure and response, which assumes reciprocity between the inputs and outputs
- Artificial Neural Networks, seeking to identify a non-parametric, inverse relationship between the outputs and the input.

This work concentrates on the latter approach, in particular seeking development of the efficiency and robustness of ANN training.

1.3 Challenges to Laboratory Technique Development and Subsequent Production Deployment of Indirect Cylinder Pressure Reconstruction

This work is limited to reconstruction of cylinder pressure measured under laboratory conditions. While this is a sensible initial approach to acquire high quality and repeatable data (hence minimising the difficulties of ANN training), it is prudent to understand that ultimately the deployment to production engine use will present additional challenges.

The laboratory test arrangement reported in this thesis uses an inline 3 cylinder (I3) Direct Injection Spark Ignition (DISI) engine. This configuration has the advantage that combustion events between cylinders are well spaced in the crank angle domain, with negligible overlap of the pressure signals. This helps to simplify the inverse model the ANN is attempting to fit. With the current direction of engine downsizing and a number of 3-cylinder engines now appearing in the market, this is timely and an appropriate configuration to study. However, any success of the method may require further work to apply to 6 or 8 cylinder engines where firing separations are much smaller.

The driveline configuration of the test engine is also significantly simpler than that to be found in a vehicle. The engine flywheel is for example connected directly to the dynamometer via a flexible coupling, without a gearbox or clutch installed, and with minimum ancillary components driven from the crankshaft nose. For a production engine installed in a vehicle, many more variables are introduced which may influence both crank dynamics and engine vibration. In fact, all of the following factors will have an influence on crankshaft acceleration:

- Clutch stiffness, or dual mass flywheel characteristics
- Selected gear ratio, and resulting changes in referred inertia from other parts of the driveline
- Variable ancillary components driven from the crankshaft, including air conditioning compressor, clutched supercharger etc.
- Road inputs to the vehicle suspension and the driveline where the torsional response of the driveline (e.g. tyre slip) may result in engine vibration signals that are not sourced from the combustion pressure or other engine sourced inputs that have been successfully rejected by the method

All of the above points acknowledge that production implementation will require specific development for engines mounted in vehicles.

Development of the method on a single reference engine dataset will leave a remaining adaptation step to ensure that production variability, including variations though engine life, can be accommodated.

1.4 A Literature Survey of Cylinder Pressure Reconstruction by Indirect Means, and Recent Recurrent ANN Training Developments

Cylinder pressure reconstruction has been subject of research for many years, with publications commencing in the 1980s, and continuing in significant numbers to the present day. The majority of publications use either crankshaft kinematics or engine structural vibration as inputs, and a variety of modelling methods are used to reconstruct cylinder pressure,

including inversion of physical models, observer methods and increasingly, artificial neural networks.

The following literature review is therefore simply structured into three categories: first by considering the literature addressing cylinder pressure reconstruction for both diesel and gasoline using crank kinematics, then by focusing on reconstruction using engine block related vibration signals, again for both diesel and gasoline, finally papers offering recent developments in recurrent neural network training methods are reviewed

Cylinder pressure reconstruction using crank kinematic inputs

[Rizzoni G. 1989] One of the earliest paper introducing crankshaft speed fluctuation as a route to estimating indicated torque, the various sources of torque input to the crankshaft are defined, including gas pressures, inertia forces and friction torques, and an electrical circuit analogy developed as a lumped parameter model of the crankshaft rotational dynamics. Experimental data was taken from a 1.5 litre gasoline I4 engine instrumented for 1 cylinder pressure and crankshaft speed with both an optical encoder and a flywheel teeth probe, at speeds from 1500 rev/min to 3500°rev/min. Although cylinder pressure reconstruction is not specifically demonstrated, the indicated torque measures given are directly related, and show good correlation.

[Jacob et al.1999] and [Gu et al.1999] The first of these papers (Part 1) proposes an RBF network to reconstruct pressure from instantaneous crankshaft velocity sampled in the angle domain. The initial physical model is described, and the steps of creating a RBF network including selection of centres, regularisation and model order selection are described. The second of the paper series (Part 2) applies the approach to an I4 2.5 litre diesel engine instrumented in one cylinder for pressure and with flywheel teeth instrumentation for crankshaft speed. Data was acquired over test conditions with speeds between 1000 rev/min and 2000 rev/min, and with loads between 20 Nm and 60 Nm. Pressure data was sampled over a window of 160°, crank speed a window of 180° around TDC, with separate training and validation datasets acquired. The RBF network was configured with radii

ranging from 1000 to 10000, and used 100 centres. P_{max} was reconstructed typically within 5%, and θ_{max} within ±2°.

[Haskara I. and Mianzo l. 2001] An on-line estimator for indicated torque and individual cylinder pressure is proposed, using crankshaft speed and position as inputs to a second order sliding mode differentiator for real time differentiation to estimate instantaneous torque. Crankshaft speed is also used as input to a friction torque calculation which is then subtracted from the total torque. Combustion pressure estimation is approached with the estimate indicated torque injected to the combustion dynamic model, which is used only on the power stroke. Simulations are shown for a Simulink model of a 2 litre I4 gasoline engine, and the P_{max} from the combustion pressure estimator is within approximately 5% of the Simulink prediction. The point is made that the system is unobservable at TDC.

[Hamedovi et al. 2005] This publication offers an interesting approach by combining a model based on crank kinematics with measurement of a single cylinder's pressure signal in a multi-cylinder engine. A parametric pressure model is proposed, where pressure is decomposed to a compression and a combustion component, with the compression component assumed equal over all cylinders within any one cycle, and both components decomposed from the reference measurement. The compression pressure is also used to calculate instantaneous torque through the relevant angle in conjunction with engine speed and its derivative. The paper considers an I4 gasoline DISI engine, and reconstructs pressure for a single operating point with P_{max} predictions within approximately 5-10% and θ_{max} within approximately ±5°.

[Johnsson 2006] A complex valued radial basis function trained with a version of recursive hybrid learning is employed to reconstruct pressure from structural vibration and crankshaft speed signals, using the Fourier transforms of these signals as inputs. Experimental data was generated from a 9 litre, 6 cylinder ethanol fuelled diesel engine, instrumented for pressure in cylinder no. 1, vibration on a cylinder head bolt chosen as dominated by the forcing function of interest (i.e. cylinder

pressure) and crankshaft speed by an optical encoder. Data was sampled in the angle domain, at speeds between 800 rev/min and 2000 rev/min with load conditions between 10% and 90%. The best compromise of RBF network was found with 39 centres and 6000 width. The network's reconstruction RMS error for P_{max} was 2.9% (verification) and 3.4% (validation), with RMS error in θ_{max} 1.5° for the validation dataset.

[Potenza R. et al. 2007] Cylinder pressure reconstruction is approached with a NARX architecture neural network. Inputs to the network are crankshaft position, crankshaft acceleration, delayed crank acceleration and delayed, predicted cylinder pressures. The output of the network is cylinder pressure, with the network providing the inverse model of the crank-train dynamics. The network was constructed with 8 hidden layer neurons. The authors note the recurrent feedback of predicted pressure make network training demanding. Two training methods are demonstrated, back propagation through time (BPTT), and extended Kalman Filtering (EKF). Experimental data is gathered from a 1.12 litre I3 gasoline DISI engine. The network predictions were good, with RMS errors below 2 %, but with some larger local error, particularly close to TDC. The EKF training was found to be more efficient than BPTT, but both suffered from long training durations.

[Saraswati S. and Chand S. 2010] The authors describe a neural network approach to cylinder pressure reconstruction using crankshaft speed, modelled motored pressures, spark advance and equivalence ratio as inputs. A recurrent neural network is described, with NARX architecture. A two zone heat release model is used to generate cylinder pressures for training, but the training method is not detailed. The network structure is varied, and optimum values for the number of inputs are identified, recurrent outputs fed back to the input is found to be optimum at 3, the number of delayed inputs for engine speed is found to be optimum at 7, and the number of delayed values for motored pressure is found to be 7. Some experimental data is also compared to network predictions with maximum pressure errors of approximately 5-10%.

[Al-Durra A. et al. 2011] An estimation method is proposed that uses crankshaft speed to predict the pressure trace for individual cylinders, with other sensor inputs including fuel mass flow rate and intake manifold pressure also used in the calculation. An energy conservation approach is used to describe cylinder pressures between intake and exhaust valve closure, and pressure during charge exchange is assumed constant. A sliding mode observer is implemented for cylinder pressure estimation, with instantaneous crankshaft speed as the only measured variable, the sliding mode gain designed as a function of crank angle to mitigate control chattering. An augmented estimator is also designed to improve pressure predictions during the charge exchange parts of the cycle. The pressure estimator was tested against data from a 2.5 litre I4 diesel engine, and performed with errors in $P_{max} \approx 2$ % and in θ_{max} ±2°.

[Liu F. et al. 2012] A review of reconstruction literature leads to a definition of a physical crankshaft dynamic model, and the equations relating instantaneous crankshaft speed change to total torque acting on the crankshaft. Signal processing parameters around the treatment of engine speed are considered, with low pass FIR filtering at 18th harmonic order cut-off used to treat the raw signal. Torque estimate from the speed signal is compared to measured data with good correlation, with a warning regarding the rigid body assumption becoming less valid at higher speeds. A thermodynamic model is shown for the compression part of the cylinder stroke, and the gas torque component extracted from the total instantaneous torque equations. The equation relating gas torque to pressure is then inverted, and the discontinuity close to TDC is dealt with by an interpolation method between the valid parts of the predicted curve, and the Wiebe function is shown as a route to this calculation. The experimental engine used was a 2.0 litre I4 diesel engine and modelling errors of P_{max} of 2.3 % - 11.2 % and θ_{max} of -0.4° to 4.4° are quoted.

[Taglialatela et al. 2013] Seeks to determine peak pressure value and location using a neural network approach, both crankshaft speed variation and crankshaft acceleration were inputs to a multi-layer perceptron

with 30 hidden neuron employing *tanh* activation functions. The outputs of the ANN were peak pressure magnitude (P_{max}) and peak pressure location (θ_{max}). Experimental data was generated from a single cylinder gasoline research engine, using in-cylinder pressure transducer and optical encoder for instantaneous crankshaft speed measurement. Data was acquired in the angle domain, with a 0.1° resolution, at 200 rev/min steps over the range 1000-2000 rev/min, all speed conditions being at wide open throttle, with varying intake boost pressures. The reported mean root mean square errors of P_{max} were between 4.1% and 8.0%, with θ_{max} errors of 1.4° to 5.2°, with a generally sound pattern of following high pressures with high prediction etc..

Cylinder pressure reconstruction using vibration inputs

[Gao Y. and Randall R.B. 1999] The authors identify the convolution relationship between an input and output of a linear system, and the equivalent product relationship in the frequency domain. This is extended to the inverse filter concept using the reciprocal or the frequency response function (FRF) between two signals. The difficulty of obtaining, and the variation of FRFs is acknowledged, and the averaging of FRF across operating condition of an engine is recommended. Cepstral smoothing to reduce the influence of FRF variability is introduced, smoothing being achieved by low pass liftering (multiplication in quefrency domain) of the corresponding complex cepstra. The time domain smoothing method is discussed in detail, its implementation simplicity is noted, requiring one windowing and 2 Fourier transform operations. It is applied to experimental data from a single cylinder 4-stroke diesel, and compared with direct and cepstral smoothing techniques.

[Du H. et al. 2001] Cylinder pressure is reconstructed from vibration signals using a radial basis function neural network. The authors identify the complexity of the vibration signal, and the difficulty of identifying a simple model for the transfer function, and identify a feed forward neural network with radial basis function hidden neurons as a candidate for the model. Experimental data is from a 2 cylinder 4-stroke diesel engine instrumented with a cylinder pressure transducer in 1 cylinder and a knock sensor for vibration, TDC and crank angle signals. 4 different speeds and 3 different

loads make up the operating conditions tested. Vibration data is transformed into a power spectrum and the data below 15kHz used as input to the network. Specific errors of prediction are graphed but not enumerated.

[Antoni J. et al. 2002] The authors seek to reconstruct cylinder pressure from structural vibration measurements using an optimal inverse filter. A previous part of the paper by the authors sets out principles of cyclostationary processes applied to condition monitoring of engines. Potential problems are identified, being concern that the vibration signal contains enough information to describe the cylinder pressure, and whether the combustion contribution can be isolated within the vibration signal. The problem that cylinder pressure has much low frequency energy while vibration signatures tend to have little low frequency energy due to stiff engine structures is acknowledged, as is the challenge of additional noise contributions from other excitation sources. Careful placement of the vibration transducer is advised, and the cylinder head bolt is identified as preferred, the location having low sensitivity to piston slap in particular. A linear relationship between input pressure and output vibration is assumed, and the approximation of this assumption acknowledged, and the fact that the relationship will be crank angle dependant is stated. The relative advantages of time and angle domain sampling approaches for acquiring data are discussed, with the time domain being less likely to lead to the method being highly dependent on a specific engine speed. The theoretical formalisation of the convolution issue to return the vibration signal to its source is set out, and the problem is seen as essentially the identification of some inverse filter, with the direct identification of this inverse filter being preferred over inversion of the forward filter. A practical implementation is derived under the assumption of cyclostationarity, employing cyclic averaging and the use of the method has 2 phases – identification phase to model the inverse input-output relationship and deconvolution phase to utilise this inverse model

Experimental data was gathered from a 4 cylinder 2-stroke diesel with one pressure transducer and 2 accelerometers on cylinder head bolts. An

optimal inverse, periodically varying filter was estimated using a filter bank structure, and now weighting function to isolate combustion signatures in the vibration signal was necessary. No error figures are given, but the reconstruction comparisons with measured pressures are good, with maximum pressure errors around TDC of approximately 5-10% and a good match of the curve shape, including retarded start of combustion beyond TDC.

[Villarino R. and Böhme J.F. 2003] The approach for this publication relies on the assumption that the structure-borne sound (vibration) of the engine structure consists of a superposition of different components, one for each cylinder, being filtered versions of the original pressure signals. A combination of three terms is considered for each component – a filtered version of past sound samples, a cylinder dependant time-variant filtered version of the pressure component and a noise term. The periodic movement of the piston leads to the use of a time varying transfer function. A method of sound signal decomposition is explained, and some of the difficulties are identified with expectation–maximization (EM) proposed as an alternative, decomposing the original multi-dimensional problem to a set of simpler ones solved in parallel. Experimental data is taken from 1.8 litre turbocharged spark ignition engine, with spark plug pressure transducers in each cylinder, four accelerometers mounted on the cylinder block and a crank angle sensor. Data from a single cylinder and a single test point are shown, when driven by the vibration signals the error prediction of P_{max} appear to range over approximately 25%, but θ_{max} is predicted with a mean of 0.04° and a standard deviation of 4.8°.

[El-Ghamry M. et al. 2004] The input to cylinder pressure reconstruction in this case is acoustic emission (AE), identified in another of the author's publications as high frequency waves travelling through or over a solid. Both raw and RMS AE signatures are considered, acquired from test engines (1 a large 2-stroke marine diesel, the other a small 4-stroke diesel) via sensors on one of the cylinders. An advantage of AE is claimed as its high signal to noise ratio. RMS AE signals are treated first by low pass

filtering <2.5 kHz, raw AE signals are considered unsuitable for use directly in the frequency domain, and are transformed using an envelope function, the result being similar to the RMS AE. Raw AE (without enveloping) was used for time domain modelling, though could not be used to model the compression phase of the cylinder pressure, which was modelled with polynomials fitted to test data for the various loading conditions. The combustion and expansion phases are modelled with raw AE using an auto-regressive moving average technique. Complex cepstral analysis is utilised on the RMS AE signal, and is used to model the whole combustion interval, without curve fitting on the compression phase. A comparison is made for reconstruction using AE with that from acceleration, with evidence that AE offers benefits. Pressure reconstruction errors are quoted as being typically 7%. The author makes an interesting comment that attempting to use the same technique on a large gasoline engine saw significantly larger reconstruction errors, blamed on lower pressures resulting in a less consistent RMS AE pattern, and a resulting decreased stability of the cepstrum model.

[Yong X. et al. 2010] This paper describes the application of a neural network to the reconstruction of cylinder pressure from cylinder head vibration. The difficulties of the physical phenomena are discussed, and the conclusion that the non-linear prediction capabilities of a neural network may be well suited to the problem is drawn. The specific architecture of the network used is not described in detail but the input layer / hidden layer / output layer description suggests a feed forward network, trained by back-propagation. Experimental data was taken from a 6 cylinder 4-stroke diesel engine instrumented for cylinder pressure on cylinder 6, cylinder head vibration and a TDC pulse. Data was acquired from 6 different load conditions from 70° BTDC to 70° ATDC. Pressure reconstruction errors were approximately $P_{max} \approx 2$ % and in θ_{max} ±3.5°.

[Bizon K. et al. 2011] The authors' objective is to build an efficient and robust radial basis function ANN to reconstruct cylinder pressure using an accelerometer signal as the input. The experimental set-up used a single

cylinder diesel engine instrumented for cylinder pressure and with a low cost accelerometer mounted on the cylinder block just below the head gasket. Data were sampled in the crank angle domain at 0.5° intervals. The neural network model was a feed forward architecture with RBF activation function in the single hidden layer. The network parameters in terms of the number of centres was established, and found that 50 neurons offered no significant improvement over using 5 neurons. The network was then trained using data from different operating conditions – 100 cycles presented to the network for training for each of 6 different speed / load / fuel type combinations. RMS prediction errors were < 2.7% for P_{max} and <1.5° for $\theta_{max.}$

Artificial neural network training

[Mandic D.P. et al. 2001] A fully adaptive normalised, nonlinear gradient descent (FANNGD) algorithm is described, for the online adaptation of non-linear neural filters. As an evolution of a previous NNGD algorithm, an adaptive learning rate for the NNGD is developed according to a gradient based method. The new method is tested against NNGD in a Monte Carlo simulation and seen to outperform the preceding method. Initial conditions are recognised as crucial, and performance of the filter dependant on the number of tapped delays. Convergence boundaries are also discussed.

[Rubio J.J. et al. 2008] A neural network training method is proposed based on the Optimal Bounding Ellipsoid (OBE), with potential for faster convergence than gradient based methods. The OBE's application to ANNs is described, and the method of updating weights is derived, a key feature possibly being that the learning rate is not a constant, but a matrix that changes over time. Lyapunov stability theory is employed to confirm the method's convergence. The proposed method is applied to a nonlinear system identification task, and is shown to perform better than back-propagation.

[Song Q. et al. 2008] An adaptive gradient descent method is designed for the training of recurrent multi-input, single-output (MISO) ANNs. A weight update algorithm is developed using standard on-line back propagation (BP) and real time recurrent learning (RTRL) according to required convergence

and stability conditions. An extended recurrent gradient is defined, and an adaptive learning rate similar to the classical dead zone approach is employed. Methods for update of both output and hidden layers of the network are derived and the stability of the algorithm is discussed in detail via Lyapunov and conic sector theories. The method is applied to experimental simulations and shown to outperform both RTRL and normalised RTRL approaches.

After having reviewed the literature, a position has been reached where it is evident that despite the considerable body of literature available, there remain a number of important areas for which further research is needed. In particular, the accuracy of gasoline engine cylinder pressure reconstruction is generally still not adequate for combustion control purposes. For example, the literature shows that existing methods of reconstruction for gasoline engine are not capable of consistently predicting P_{max} within 4% of target, and θ_{max} within ± 2°. Furthermore, the literature shows that neural network based cylinder pressure reconstruction appears to be a very appropriate route but robust training of recurrent networks remains a considerable challenge. There are also issues which evidently need further investigation, concerning the consistency of real engine data collection, it's processing, and the need to address issues of sampling and frequency content – all of which have an important impact on neural network based reconstruction. It is from this motivation that the following objectives of study can be set:

1.5 Thesis Objectives

The overall objectives of this thesis are:

- Acquisition and reduction of a reference engine dataset for application to ANN training for cylinder pressure reconstruction, including the details of necessary signal processing including sample rates, filtering, calibration and synchronisation of signals
- Confirmation of the suitability of a NARX network structure for cylinder pressure reconstruction purposes
- Identification of up to date recurrent ANN training methods and implementation of the most suitable using the reference engine dataset

- To understand the importance and sensitivities to training method parameters to achieve reliable and efficient training
- Demonstration of the chosen training method's capability for robust cylinder pressure reconstruction and ability to adapt for engine variability

1.6 Thesis Structure

A summary of the thesis structure is as follows:

Chapter 1 introduces the reasons for interest in reconstructing cylinder pressure for internal combustion engines, the possible approaches and challenges to achieving a successful implementation and the previous work conducted in the field.

Chapter 2 discusses the mechanics of crank kinematics and identifies specific problems of indirect cylinder pressure reconstruction and the desired inputs to the potential inverse models.

Chapter 3 introduces neural network based system identification and inverse modelling, discussing the available network architectures and the methods available for training them.

Chapter 4 describes the engine test facilities, sensors and data acquisition systems used to generate the datasets used for training and validation of neural networks.

Chapter 5 details the methods used to process the signals acquired from the running engine to achieve high quality data input to the neural network training exercises.

Chapter 6 shows the results of NARX ANN structures using feed forward and teacher forced training.

Chapter 7 details the work undertaken to implement and utilise a recurrent training strategy using the Robust Adaptive Gradient Descent (RAGD)

method applied to simplified modelled data and to real data from the operating engine.

Chapter 8 draws together conclusions from the work, and considers where future efforts could be targeted.

Chapter Two

2. INDIRECT CYLINDER PRESSURE RECONSTRUCTION – THE PROBLEM

2.1 Overview

The external measurements that form the basis of this thesis are crank kinematic data, and cylinder block vibration. Significant components of both signals will be sourced from the combustion pressure in the cylinder. However, both signals also have many other inputs to their sum, they are multi-input single-output systems as measured, and the extraction of the cylinder pressure component from the other energy sources is the challenge. Apart from the mixture of energy sources contributing, the non-cylinder pressure inputs may have contributions to the measurement signals equal or greater than the cylinder pressure source, making signal to noise ratio of the measurement a problem in itself.

Inversion of a direct physical model may be difficult, even in the absence of other inputs – for example direct inversion of the crank acceleration model becomes singular, as described in section 2.7.

2.2 Crank kinematic definitions

The instantaneous acceleration of the crankshaft can be described by the simple relationship in equation (2.1) below:

$$\alpha_r = \frac{T_r}{I_c} \tag{2.1}$$

where

α_r = resultant instantaneous crank shaft angular acceleration

T_r = resultant torque acting about the crankshaft's axis of rotation

I_c = mass moment of inertia of crankshaft about its axis of rotation

This simplified description of the dynamic system assumes that the crankshaft is internally rigid, and that the response to external inputs is inertia

controlled. Section 4.2 and section 5.7 respectively suggest that these assumptions are reasonable. However, the factors which contribute to the resultant torque T_r are multiple and complex. A number of physical phenomena contribute to the resultant torque at any instant, and most are both time and crank angle dependent.

Cylinder pressure gas force:

Gas pressure exerts a force on the piston crown and piston rings, acting axially along the cylinder bore and transferred to a torque on the crankshaft via the connecting rod. The gas pressure itself is a function of both time (the combustion event itself has time dependency) and crank angle due to the varying volume of the combustion chamber. The resolution of the piston force to a torque is crank angle dependent. Cylinder and piston wall temperatures will influence the pressure to a secondary degree, but should be effectively invariant during steady speed and load operation. Compression ratio and connecting rod length will influence pressure and torque resolution respectively, but are constant for a given engine geometry.

Reciprocating inertia forces:

The acceleration of the piston oscillating in the cylinder bore requires force acting axially along the bore. This axial force is transferred to a torque at the crankshaft via the connecting rod. Mass of the piston assembly directly influences the magnitude of the force and hence torque and is constant for a given engine. In the absence of more detailed information, 1/3 of the connecting rod mass is conventionally approximated to be a part of the reciprocating assembly, with the remaining 2/3 considered as a rotating part centred on the crank shaft big-end bearing and effectively increasing the crank's inertia. The inertia force is crank angle and crank angular velocity dependent. In a vertically mounted engine such as the Ford I3 research engine, piston assembly mass results in a small offset to the inertia force acting on the piston.

Friction forces:

Friction is present in all of the moving components comprising the cranktrain, including the crankshaft main journals, big and small end journals, piston ring

to cylinder bore interface and the piston skirt to cylinder bore interface. The resolution of the piston sourced components to crank torque are crank angle dependent, and gas pressure influences both the piston ring to cylinder bore contact forces by acting on the back of the compression rings, and skirt to bore thrust force via the thrust angle of the connecting rod, also crank angle dependent. All systems driven by the crankshaft will also contribute frictional losses, including the valvetrain, oil pump, fuel pump and front end accessories, which on the Ford I3 comprise the water pump, alternator and power steering pump. Each of those driven components will also add rotational inertia to the crankshaft, but no direct data is available for these factors.

Valvetrain:

In addition to the friction forces discussed above (sourced at cam bearings, valve stem and actuation mechanism, gear / chain / belt drives), the valve actuation also introduces torques as a result of overcoming spring and valve inertia forces. The torque seen at the crankshaft is negative (reduces the crank velocity) while opening the valves, and positive while the spring recovers during valve closing. These torques are crank angle dependent (camshaft drive is fixed for the Ford I3 engine i.e. no variable phasing or lift) and also have crank velocity dependent components (valve acceleration is a function of crank velocity, spring deflection forces are not).

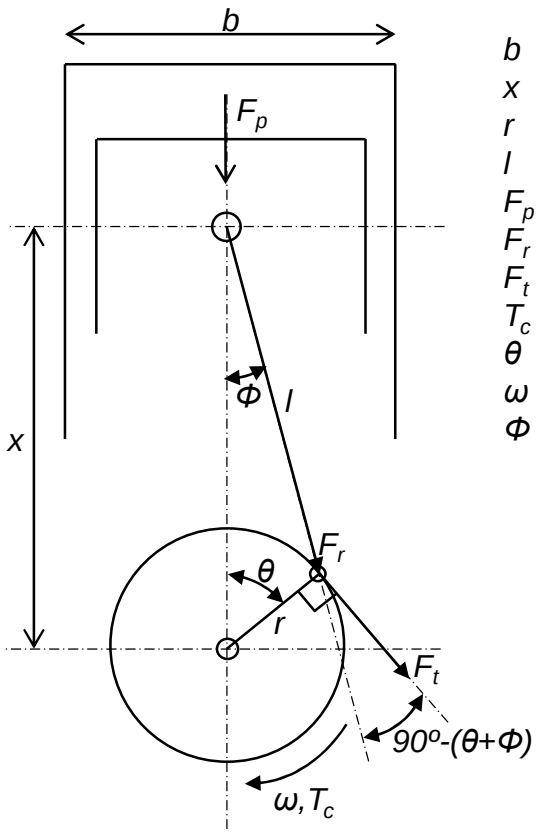
Additional Inputs:

A number of other forces act on the crank, including inputs from the driveline. As discussed in section 4.2, for the Ford I3 installation, the driveline dynamics are shown to be decoupled, although additional inertia is added by the shaft segment between the flywheel and the torsionally flexible coupling. The engine is fitted with a crank nose torsional vibration (TV) damper which will introduce a significant torque when it is in its operational range – the data shown in section 5.7 suggests that the low speed and load test conditions used for cylinder pressure reconstruction studies in this thesis are well separated from the conditions where the crank is torsionally active, and hence it is assumed that the TV damper is not introducing significant torque.

Overall, the summation of torques acting on the crankshaft is complex. Cylinder pressure reconstruction seeks to extract a single source from the resultant response of many inputs, and any simplification of the information presented to the reconstruction method will be of benefit.

2.3 Crank Kinematic Governing Equations

A crank-slider mechanism sits at the heart of the reciprocating internal combustion engine. The mechanism has complex kinematics that have an important impact on the relationship between the primary driving force of the engine (in-cylinder pressure), and the resulting instantaneous displacement, velocity and acceleration of the crankshaft itself as measured by the instrumentation described in section 4.3. Figure 2.1 shows the geometry and describing symbols of the crank-slider mechanism.



b = cylinder bore diameter
x = piston axial displacement
r = crank throw (half piston stroke)
l = connecting rod length
F_p = force on piston
F_r = force in connecting rod
F_t = force tangential to crankshaft axis
T_c = torque on crankshaft
θ = crank rotation from TDC
ω = crank angular velocity
Φ = connecting rod angle to bore axis

Figure 2.1 – Diagram and symbols of crank-slider mechanism

The governing equations relating crank position, velocity and acceleration with cylinder pressure are now derived.

2.3.1 Resolution of Axial Piston Force to Crankshaft Torque

from trigonometry (referring to Figure 2.1):

$$l \sin\phi = r \sin\theta$$

$$\sin\phi = \frac{r}{l}\sin\theta$$

$$\cos\phi = \sqrt{1-\frac{r^2}{l^2}\sin^2\theta} \tag{2.2}$$

the force in the connecting rod, F_r is:

$$F_r = \frac{F_p}{\cos\phi} \tag{2.3}$$

and the force tangential to the crankshaft F_t is

$$F_t = F_r \cos\left(90-(\theta+\phi)\right)$$
$$F_t = F_r \sin\left(\theta+\phi\right) \tag{2.4}$$

Substituting for F_r from equation (2.3) into equation (2.4)

$$F_t = F_p \frac{\sin\left(\theta+\phi\right)}{\cos\phi}$$

i.e.

$$F_t = \frac{F_p}{\cos\phi}\left(\sin\phi\cos\theta+\sin\theta\cos\phi\right)$$

this becomes:

$$F_t = \frac{F_p}{\sqrt{1-\frac{r^2}{l^2}\sin^2\theta}}\left(\frac{r}{l}\sin\theta\cos\theta+\sin\theta\sqrt{1-\frac{r^2}{l^2}\sin^2\theta}\right)$$

and finally:

$$F_t = F_p \sin\theta\left(1+\frac{r}{l}\frac{\cos\theta}{\sqrt{1-\frac{r^2}{l^2}\sin^2\theta}}\right) \tag{2.5}$$

For engines where $\frac{r^2}{l^2}$ is small (for the Ford I3 engine it equals 0.078), equation (2.5) is commonly simplified to:

$$F_t \approx F_p \sin\theta\left(1+\frac{r}{l}\cos\theta\right) \tag{2.6}$$

And the torque acting on the crankshaft T_p due a force F_p on the piston is:

$$T_p \approx F_p \, \mathrm{r} \sin\theta \left(1 + \frac{r}{l}\cos\theta\right) \tag{2.7}$$

2.3.2 Gas Pressure Force Acting on the Piston

The force exerted on the piston by gas pressure F_g in the cylinder is simply the gas pressure P_g times the cross section area of the cylinder bore, i.e.:

$$F_g = P_g \pi \frac{b^2}{4} \tag{2.8}$$

Substituting the gas force from equation (2.8) into equation (2.7) gives the torque on the crankshaft sourced from the gas pressure T_g shown in equation (2.9):

$$T_g \approx P_g \pi \frac{b^2}{4} \mathrm{r} \sin\theta \left(1 + \frac{r}{l}\cos\theta\right) \tag{2.9}$$

Figure 2.2 shows an example of the gas pressure sourced torque acting on the crankshaft for the test point at 2000 rev/min and 10 Nm, using equation (2.9).

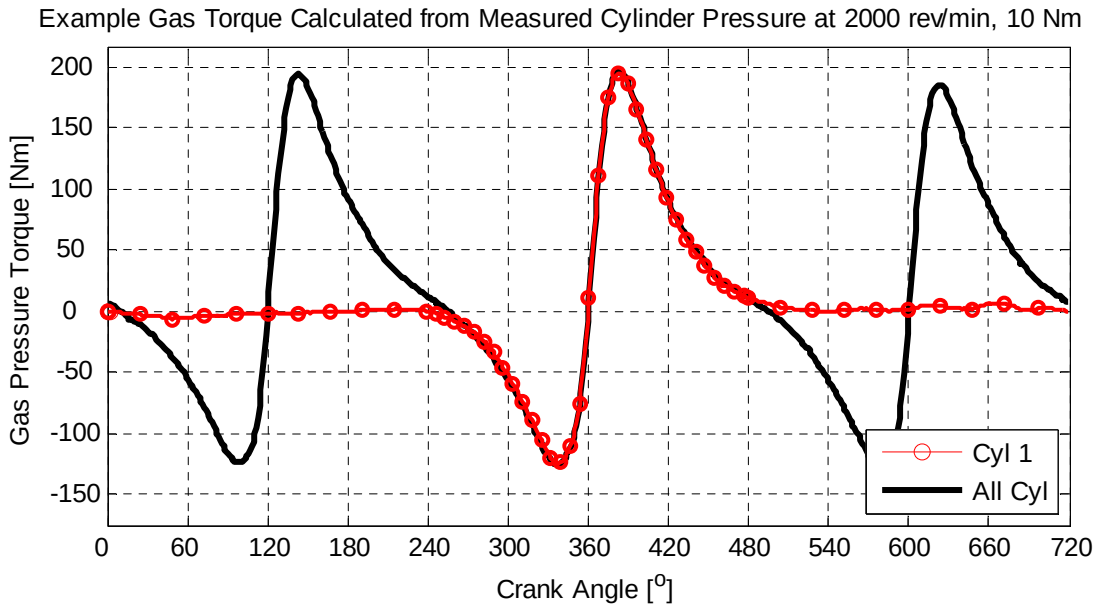


Figure 2.2 – Example gas torque calculated from measured cylinder pressure; for 2000 rev/min, 10 Nm test point
Red: Cylinder No. 1 contribution
Black: All 3 cylinders summed

The torque sourced from a single cylinder (cylinder number 1) is shown in red and the black line shows the summed contributions from all 3 cylinders, this being the total gas torque acting on the crankshaft.

2.3.3 Inertia Force Acting on the Piston

The primary motion of the piston is constrained by the cylinder bore (secondary motion is beyond the interest of this thesis), and the effective reciprocating mass is seen at the point of connection to the crank shaft link – the connecting rod. Hence, we can take the location of the connecting rod small end centre for piston position [Stone, 1999]:

$$x = r\cos\theta + l\cos\phi \tag{2.10}$$

substituting equation (2.2) into equation (2.10) and re-arranging gives:

$$x = r\left(\cos\theta + \frac{l}{r}\sqrt{1-\frac{r^2}{l^2}\sin^2\theta}\right) \tag{2.11}$$

the square root term in equation (2.11) may be expanded by the binomial theorem, i.e.:

$$x = r\left(\cos\theta + \frac{l}{r}\left[1-\tfrac{1}{2}\frac{r^2}{l^2}\sin^2\theta-\tfrac{1}{8}\frac{r^4}{l^4}\sin^4\theta+\ldots\right]\right) \tag{2.12}$$

and the powers of *sin θ* expressed as equivalent multiple angles:

$$\sin^2\theta = \tfrac{1}{2}-\tfrac{1}{2}\cos 2\theta \tag{2.13}$$

$$\sin^4\theta = \tfrac{3}{8}-\tfrac{1}{2}\cos 2\theta-\tfrac{1}{8}\cos 4\theta \tag{2.14}$$

substituting the multiple angles from equations (2.13) and (2.14) into equation (2.12) gives:

$$x = r\left(\cos\theta + \frac{l}{r}\left[1-\tfrac{1}{2}\frac{r^2}{l^2}\left(\tfrac{1}{2}-\tfrac{1}{2}\cos 2\theta\right)-\tfrac{1}{8}\frac{r^4}{l^4}\left(\tfrac{3}{8}-\tfrac{1}{2}\cos 2\theta-\tfrac{1}{8}\cos 4\theta\right)+\ldots\right]\right)$$

The $\frac{r^4}{l^4}$ term will be small and may be ignored, reducing the piston position *x* to:

$$x \approx r\left(\cos\theta + \frac{l}{r}\left[1-\tfrac{1}{4}\frac{r^2}{l^2}\left(1-\cos 2\theta\right)\right]\right) \tag{2.15}$$

Equation (2.15) can then be differentiated once to give piston velocity, and twice to give piston acceleration:

$$\dot{x} \approx -r\omega\left(\sin\theta + \tfrac{1}{2}\frac{r}{l}\sin 2\theta\right) \tag{2.16}$$

$$\ddot{x} \approx -r\omega^2\left(\cos\theta + \frac{r}{l}\cos 2\theta\right) \tag{2.17}$$

Which, given a total reciprocating mass m_r (equal to piston assembly mass plus 1/3 connecting rod mass) results in an axial inertial force F_i (including piston assembly weight for a vertically mounted engine) on the piston of:

$$F_i \approx (9.81\mathrm{m}_r) - m_r r\omega^2\left(\cos\theta + \frac{r}{l}\cos 2\theta\right) \tag{2.18}$$

Combining equations (2.7) and (2.18), to transform the inertia force acting on the piston to the tangential force acting at the crank pin and multiplying by the crank throw, the inertial torque T_i on the crankshaft becomes:

$$T_i \approx -m_r r\omega^2\left(\cos\theta + \frac{r}{l}\cos 2\theta\right) \bullet r\sin\theta\left[1 + \frac{r}{l}\cos\theta\right] \tag{2.19}$$

Equation (2.19) can be further re-arranged to the commonly quoted form [Harris 2002], of equation (2.20)

$$T_i \approx -m_r r^2\omega^2 \sin\theta\left(\cos\theta + \frac{r}{l}\cos 2\theta\right)\left(1 + \frac{r}{l}\cos\theta\right)$$

expanding the bracket terms:

$$T_i \approx -m_r r^2\omega^2\left[\sin\theta\cos\theta + \frac{r}{l}\sin\theta\cos 2\theta + \frac{r}{l}\sin\theta\cos\theta\cos\theta + \frac{r^2}{l^2}\sin\theta\cos\theta\cos 2\theta\right]$$

expressing in multiple angles:

$$T_i \approx -m_r r^2\omega^2\left[\frac{\sin 2\theta}{2} + \frac{r}{l}\sin\theta\cos 2\theta + \frac{r}{l}\frac{\sin 2\theta}{2}\cos\theta + \frac{r^2}{l^2}\frac{\sin 2\theta}{2}\cos 2\theta\right]$$

$$T_i \approx -m_r r^2\omega^2\left[\frac{\sin 2\theta}{2} + \frac{r}{l}\frac{\sin 3\theta - \sin\theta}{2} + \frac{r}{l}\frac{\sin 3\theta + \sin\theta}{4} + \frac{r^2}{l^2}\frac{\sin 4\theta}{4}\right]$$

and gathering terms:

$$T_i \approx -m_r r^2\omega^2\left[-\frac{r}{4l}\sin\theta + \frac{1}{2}\sin 2\theta + \frac{3r}{4l}\sin 3\theta + \frac{r^2}{4l^2}\sin 4\theta\right] \tag{2.20}$$

Figure 2.3 shows calculated inertia torques over one engine cycle for the Ford I3 engine using measured data. The contributions of torques from individual cylinders are summed to show the effective torque acting on a rigid crankshaft. Summed inertia torques at this speed are relatively small – of the order of <10% of the gas torque described in section 2.3.2. Inertia torques would rise at higher engine speeds.

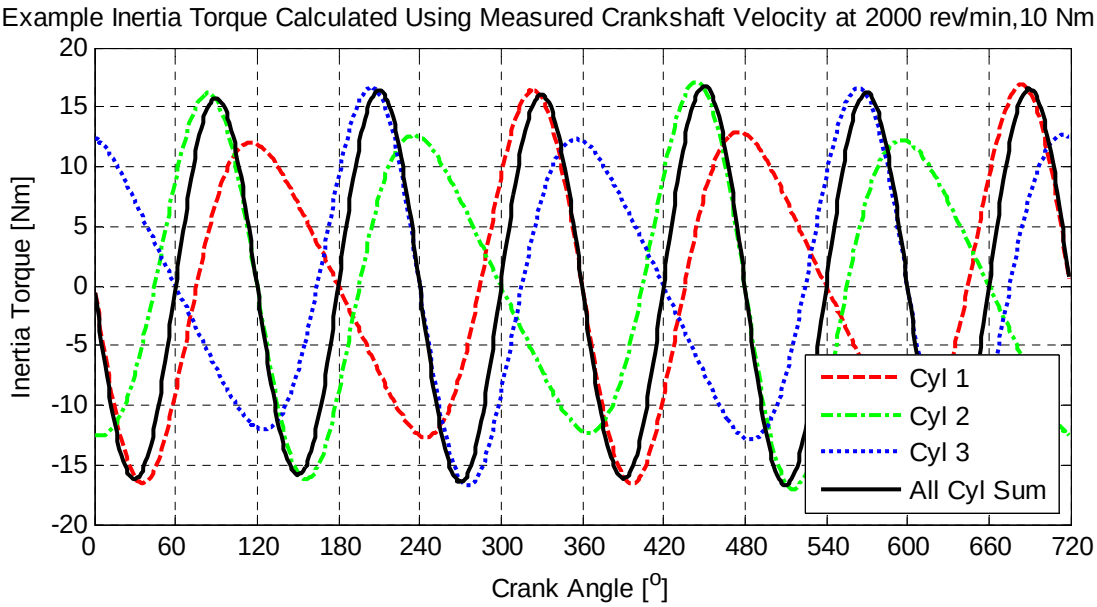


Figure 2.3 – Example inertia torque acting on crankshaft at 2000 rev/min, 10 Nm – Blue, green, red lines: Individual cylinders Black: Summation of all cylinders

2.4 Valve Train Torques

Figure 2.4 shows an example [Mufti & Priest 2012] of measured camshaft torque taken from a single cylinder gasoline research engine of 500cc per cylinder capacity. The torque curves show relatively small overall change with engine speed, an average of the curves is used to demonstrate how an example camshaft torque is used to generate a typical input to the total crank excitation.

The camshaft torques from Figure 2.4 are digitised into MATLAB, adjusted for duration and phase for each of the 3 cylinders of the Ford I3, and the

resulting typical camshaft torques over one camshaft rotation are shown in Figure 2.5.

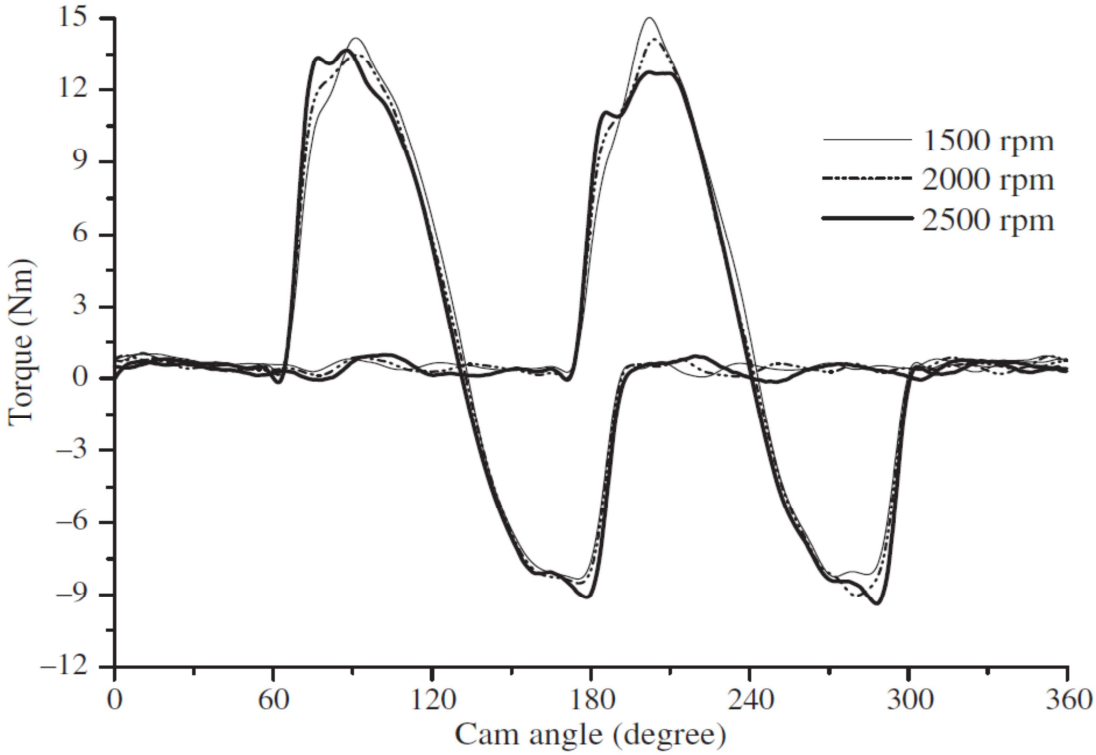


Figure 2.4 – Example gasoline engine cam torques from [Mufti & Priest 2012]

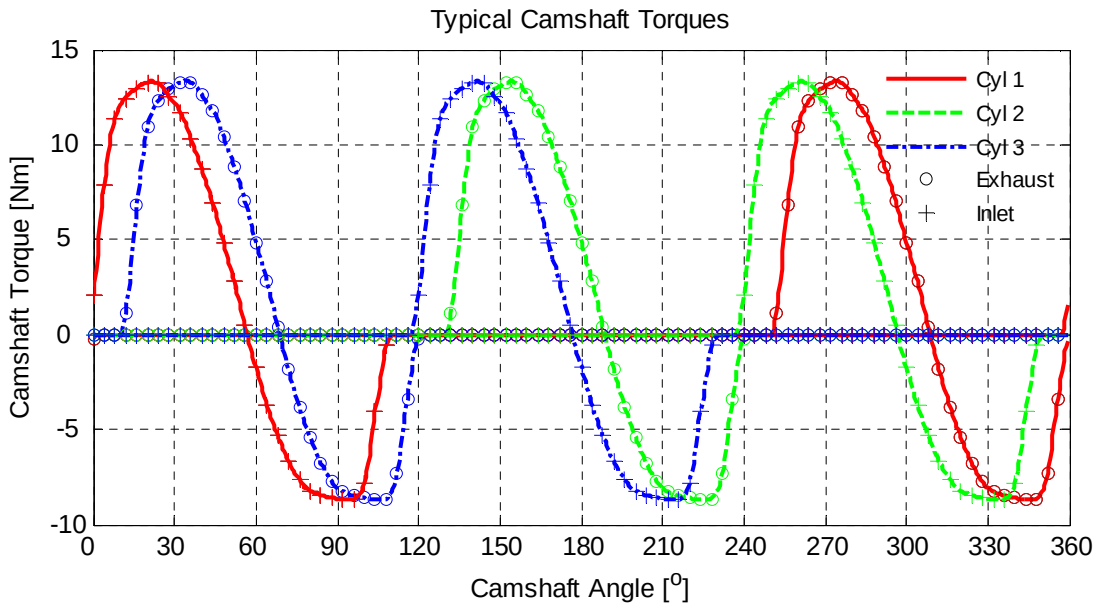


Figure 2.5 – Example camshaft torques with durations and phase adjusted appropriately for Ford I3 engine

These camshaft torques are then summed, and referred to equivalent crankshaft torques, with the assumption that torques seen at the crank sprocket of the valve train drive will be 0.5 times that seen at the camshaft end. The summed torques are shown in Figure 2.6. The peak levels of these camshaft torques are relatively low (≈6%) compared to those associated with the cylinder pressure as described in section 2.3.2.

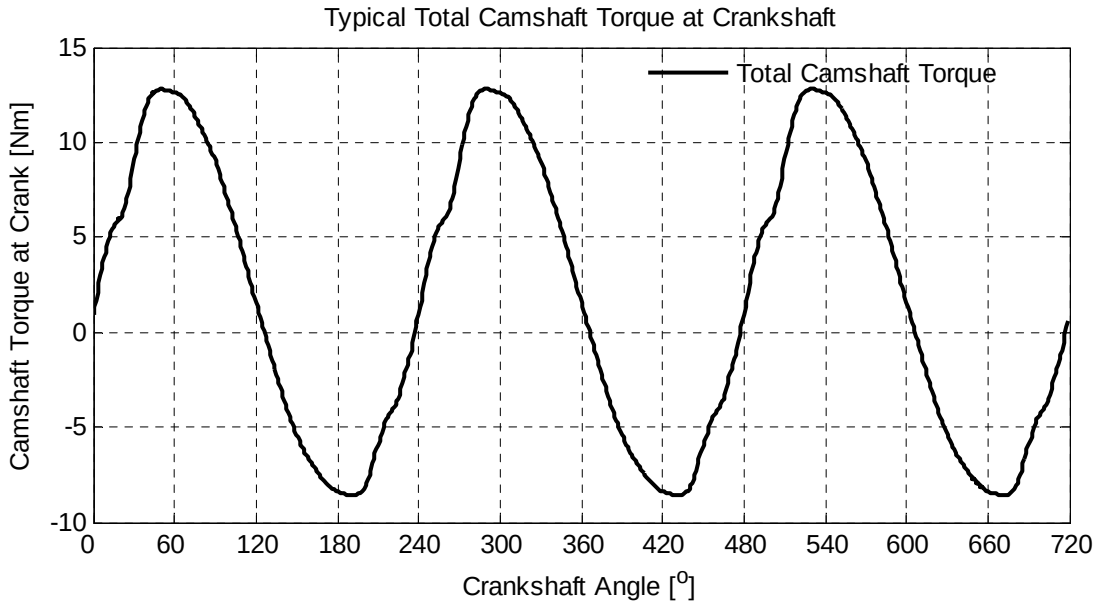


Figure 2.6 – Example camshaft torques summed and referred to the crankshaft (half torque values)

2.5 Resultant Torque on crankshaft

All of the individual torques described above in sections 2.3.2, 2.3.3 and 2.4 act on the crankshaft simultaneously. In addition, the various friction torques from every moving component in the powertrain will contribute in a crank angle and crank velocity dependant manner.

Summing the calculated torques together gives an overall resultant torque responsible for the instantaneous acceleration of the crankshaft. Figure 2.7 shows the summation of the torques approximated in the above sections, and it is interesting to compare the red (gas torque) and back (resultant torque) lines. It is clear from this example, that gas torque remains the most significant contributor to the overall resultant at this test condition, but the

peak resultant torque is 10 %-15 % lower than the peaks in gas torque. This suggests that attempting an inverted model from crank acceleration only to gas torque would result in a similar level of error.

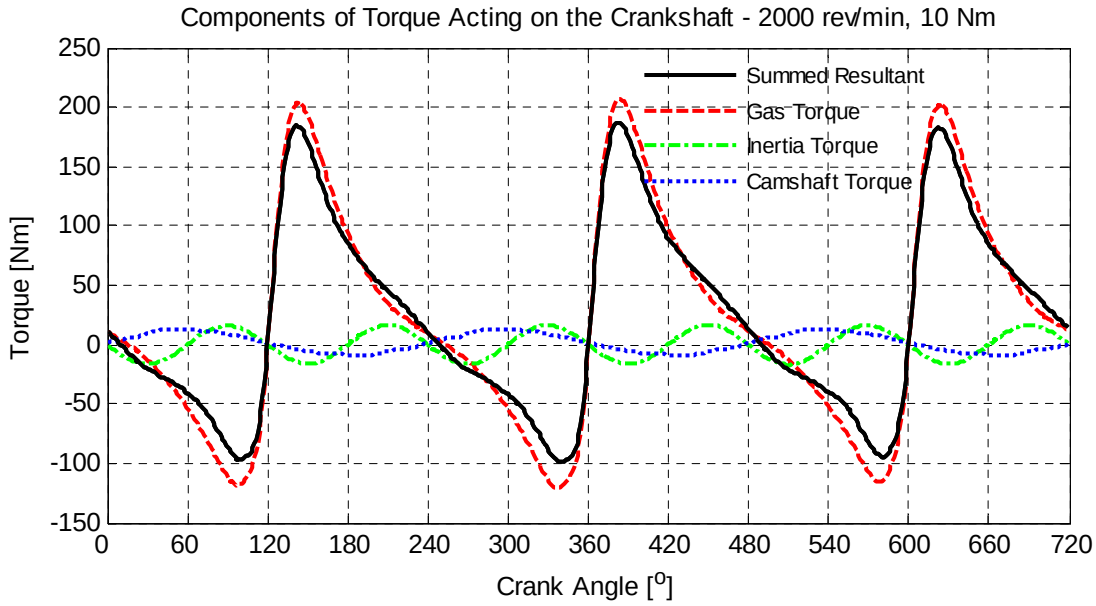


Figure 2.7 – Component and sum of torques acting on crankshaft
Red, green, blue: Gas, inertia and cam torques respectively
Black: Summed resultant torque

This model of summed resultant torque gives a mean value over the cycle of 24 Nm, while dyno torque is actually 10 Nm. The additional 14 Nm will be lost to a variety of friction sources that are not modelled above.

2.6 Modelled Crankshaft Acceleration

To improve the model a little, the additional predicted torque (14 Nm at this test point) is subtracted globally from the prediction across the cycle. This is an over-simplification as all of the contributing friction forces will have multiple dependencies within the cycle. The crank and flywheel inertia values from Table 4.1 are then summed and divided into the predicted torque to produce a prediction of crankshaft acceleration.

Figure 2.8 compares the predicted acceleration derived in the manner described above with the measured acceleration extracted from the crank nose optical encoder. While the pattern maxima and minima are broadly

consistent between predicted and measured, the errors between the curves are significant and emphasises the complex nature of the friction sources omitted from the modelled data and the difficulty of constructing a physically based model to invert crank kinematics into a cylinder pressure reconstruction.

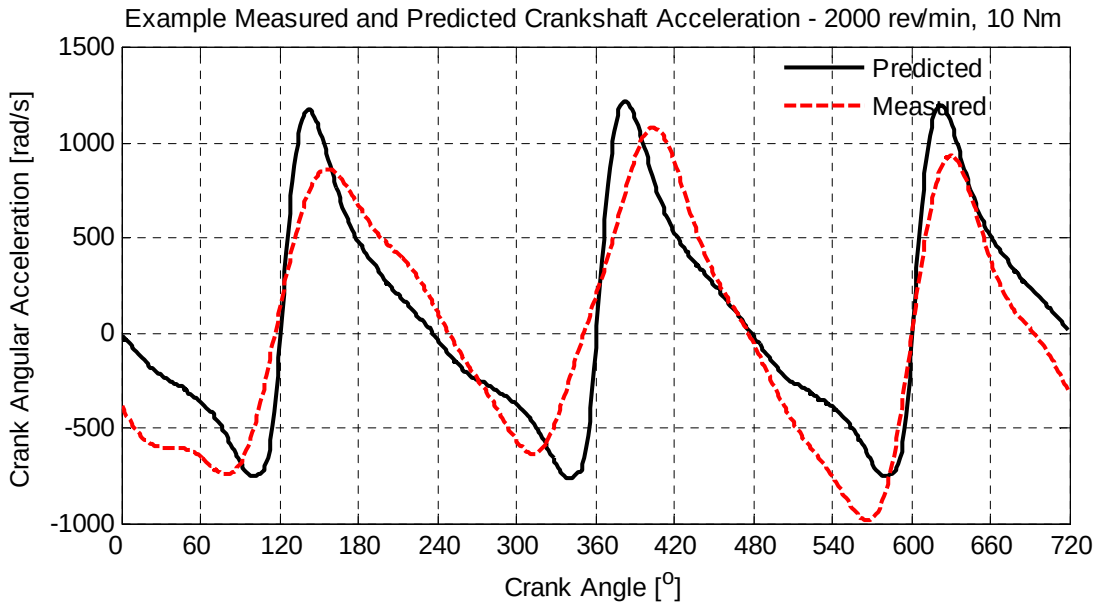


Figure 2.8 – Comparison of measured and predicted crankshaft acceleration at 2000 rev/min, 10 Nm

2.7 Inverting physical crank models

Sections 2.5 and 2.6 emphasise the multiple inputs that contribute to crankshaft resultant torque and hence the crankshaft acceleration. This alone makes inverting crank kinematics to reconstruct cylinder pressure challenging. However, the fundamental relationship between gas pressure and gas torque as described in equation (2.9), and replicated below as equation (2.21) includes a further, inherently very difficult factor.

$$T_g \approx P_g \pi \frac{b^2}{4} \mathrm{r} \sin\theta \left(1 + \frac{r}{l}\cos\theta\right) \tag{2.21}$$

Re-arranging equation (2.21) to evaluate P_g results in equation (2.22) with a term including $sin\theta$ in the denominator:

$$P_g \approx \frac{4T_g}{\pi b^2 \, \mathrm{r} \sin\theta \left(1 + \frac{r}{l}\cos\theta\right)} \tag{2.22}$$

At TDC when $\theta=0$, we also have $\sin\theta=0$ and the denominator becomes zero, the equation becomes singular and cannot be evaluated. This is a particularly difficult problem as maximum cylinder pressure occurs close to TDC and hence one of the most important features of the cylinder pressure cannot be available via direct physical model inversion.

2.8 Cylinder pressure reconstruction from engine structural vibration

The in-cylinder pressure resulting from gas compression, combustion and expansion applies significant forces to the engine structure. The gas pressure impinges directly on the structure at the cylinder head flame face, and indirectly via the piston crown, with the forces then transferred from the piston through contact with the cylinder bore and via the connecting rod and crankshaft via the big-end and main bearings. These force inputs result in a vibration response of the engine structure, which can be measured using a vibration transducer – typically an accelerometer.

If the transfer characteristics of force from gas pressure into vibration at the measurement location can be inverted, then the cylinder pressure signal could be reconstructed from the vibration data.

There are a number of challenges to this process. If the cylinder pressure were the only force input to the structure, then the response at the measurement location would have 2 primary features – a signature from the excitation energy in the pressure, and a signature related to the dynamic response of the structure, and in particular the locality of the measurement point. As we are hoping to reconstruct cylinder pressure in the time domain, the transient response of the structure is a factor. The vibration response to forces routed through the piston are not only dependent on the excitation force, but also on the changing position of the piston and the changing angle of the connecting rod, so a reconstruction method has to accommodate a time dependent transfer path as well as a time dependent force input. In a multi-cylinder engine, more than one cylinder pressure is exciting the

structure, and with different transfer paths between the inputs and a single response location, the measured output may be different even for equivalent inputs.

Of course, the cylinder pressure is not the only exciting force for the structure, many other components and events drive vibration, resulting in a multi-input, single output system that requires any reconstruction method to discriminate the inputs. Some of the inputs include piston slap – the impact of the piston on the cylinder bore as the connecting rod angle reverse close to piston top dead centre. This event is usually close to the mixture ignition point, and hence occurs within the time period of high interest for pressure reconstruction. Valvetrain opening and closing events cause high force inputs to the cylinder head – for a single cylinder engine they are well separated in the crank angle domain from the peak pressure timing. For multi-cylinder engines, valve impacts from non-firing cylinders may occur simultaneously with the combustion in the cylinder of interest (this is not a major issue for the 3 cylinder engine architecture considered in this thesis). Other force inputs include fuel injector pulses, gear impacts etc. Some of these force inputs can be discriminated by pre-processing either in the time domain if they are well enough separated from the zone of pressure interest (e.g. valve events), or in the frequency domain if they have characteristics which can be identified and filtered.

Many of the parametric approaches to inverting vibration into cylinder pressure include identification of a transfer function between the pressure and vibration, and inverting this either in the time domain by inverse filtering, or in the frequency domain using techniques such as ceptsral analysis. These methods have varying success in the literature.

2.9 Measured data required as input to invert crank and engine block models

The descriptions above of the various dynamics issues in the engine related to cylinder pressure and the external effects that may be measured lead to a choice of instrumentation suitable to gather information from an engine under operating conditions to then use for inputs to inverted models and resulting

cylinder pressure reconstruction. It is clear from the literature that crank kinematics and engine structural vibration are the favoured signals for the various methods.

Crank kinematics may be measured in a number of ways. Ultimately the signal input to the model is likely to be angular acceleration or velocity, and it is always preferable to measure the actual parameter that is required. Measurement of crankshaft angular acceleration is technically feasible (e.g. use of tangentially mounted accelerometers in the flywheel with signals telemetered to recording equipment), but challenging. Direct measurement of flywheel angular velocity is feasible by laser torsiometer, but expensive, and not available to this project. Crank angular displacement measurement is a much more proven technology and is affordable, but the data requires differentiation once to velocity and twice to acceleration. Numerical differentiation of measured signals frequently results in high levels of noise, and hence it is crucial that the angular displacement measurement is of the highest quality possible. The research seeks to achieve this with a high quality optical encoder and a novel calibration method.

Block structural vibration is more straightforward, with instrumentation quality accelerometers readily available. The knock sensor that is standard fitment to most gasoline engines is itself an accelerometer, and data from this sensor may be sufficient. This research will use both these approaches.

The acquisition of data could be sampled in either time or crank angle domain – the choice is important with respect to frequency bandwidths, and some models may require either form of data – the ability to move between the domains in data processing is desirable – this is achieved for this research as described in section 4.4.

Chapter Three

3. NEURAL NETWORK BASED SYSTEM IDENTIFICATION AND INVERSE MODELLING

This chapter seeks to introduce the relevant concepts within the field of Artificial Neural Networks (ANN), and to describe the specific architectures and structures explored by this thesis. The back-propagation learning method is described in detail, and the challenges of training recurrent networks are outlined.

3.1 The Concept of a Neural Network

It has always been clearly understood that the mammalian brain functions in an entirely different manner from a digital computer, and that the programming paradigms associated with modern computers do not replicate the information flow and decision making processes that occur in the brain. The processing paradigm of the brain is believed to be one of massively parallel, non-linear, highly interconnected elements (neurons), with re-configurable connectivity, able to re-organise its processing components to perform tasks and computation such as pattern recognition or perception many times faster than the equivalent function programmed in a digital computer. For example, the brain accomplishes perceptual recognition tasks such as identification of a familiar face in approximately 100-200ms. Estimates of the number of neurons in a human brain are approximately 10 billion, connected with 60 trillion synapses [Haykin 1999]. The human learning process is believed to result from development of the connections between neurons appropriate to matching examples of information provided by the senses.

Artificial Neural Networks (ANNs) are processing devices seeking to exploit a similar parallel, non-linear and configurable processing model as the mammalian brain, albeit on a very much smaller scale – a large ANN may consist of several thousand units, many orders of magnitude smaller (and still slower much slower) than the biological inspiration. Generally an ANN

implementation is not seeking to accurately resemble a biological system, but the concepts of learning by example and incrementally re-configuring the network such that it responds correctly are similar.

The history of ANNs is now extensive, early work by McCulloch and Pitts dates from 1943. Development paused following Minsky and Papert's 1969 book highlighting limitations of single layer perceptrons. Werbos developed the back-propagation learning method in 1974 (also developed by Rumelhart, Hinton and Williams) which remains one of the most widely applied of training approaches, and research accelerated once more through the 1980s.

The applications of ANNs are today very wide, being routinely and commercially used for a variety of pattern recognition tasks including voice and optical character recognition and automated reading of vehicle registration plates. They are also extensively used in time series prediction in financial (market and sales predictions) and engineering fields. They find use for modelling of non-linear control systems, for internet search tools and show promise in medical diagnosis.

The examples throughout this thesis simulate the parallel architectures of the various networks using normal sequential algorithms developed in MATLAB, or already available within the MATLAB Neural Network Toolbox.

3.2 A Simple Neuron Model

The fundamental processing unit of the ANN is a neuron. A model of the neuron type used throughout this thesis is shown in Figure 3.1

Some number n of inputs x are applied to the k^{th} neuron. Each input x_i is weighted by the value w_{ki} and the resulting products are summed. A bias value b_k is also included in the sum. The summed value is applied to an activation function Φ, and this function's output y_k is the k^{th} neuron's output.

The neuron can be mathematically described by equations (3.1) and (3.2):

$$u_k = b_k + \sum_{i=1}^{n} w_{ki} x_i \tag{3.1}$$

$$y_k = \phi(u_k) \tag{3.2}$$

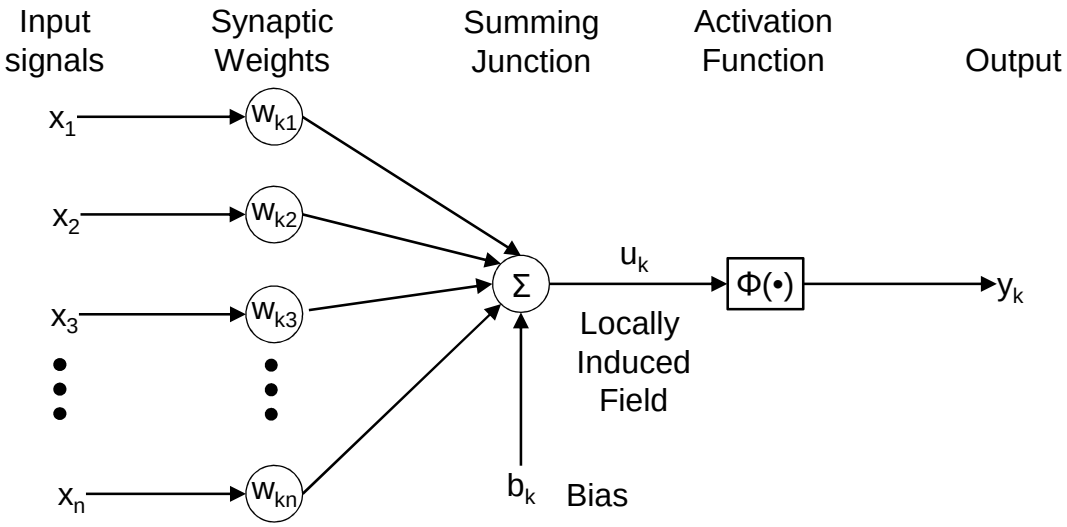


Figure 3.1 – Simple model of neuron

The value u_k feeding in to the neuron's activation function is termed the locally induced field. The bias term may be incorporated as an additional weight term by introducing a constant value input term, typically set to unity. Figure 3.2 shows this revised neuron model.

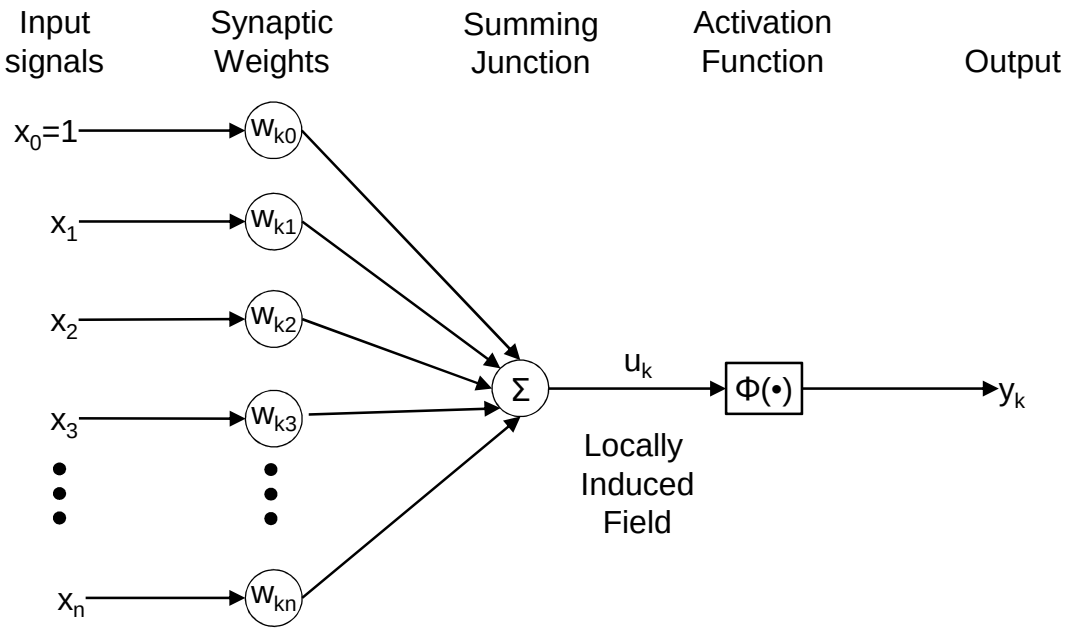


Figure 3.2 – Revised model of neuron with bias accommodated as a weighted constant input (unity)

The neuron may now be represented by equations (3.3) and (3.4) with $x_0=1$, where:

$$u_k = \sum_{i=0}^{n} w_{ki} x_i \tag{3.3}$$

and again:

$$y_k = \phi(u_k) \tag{3.4}$$

This construction means the neuron bias requires no special treatment during training, with its weight being subject to the same learning algorithm as all other inputs.

3.3 Activation Functions

The activation function Φ used in the neuron models above can take a variety of forms. These may include threshold functions as demonstrated in Figure 3.3 where the output y_k can take one of two values depending on a set threshold of the input U_k as described by equation (3.5) and (3.6).

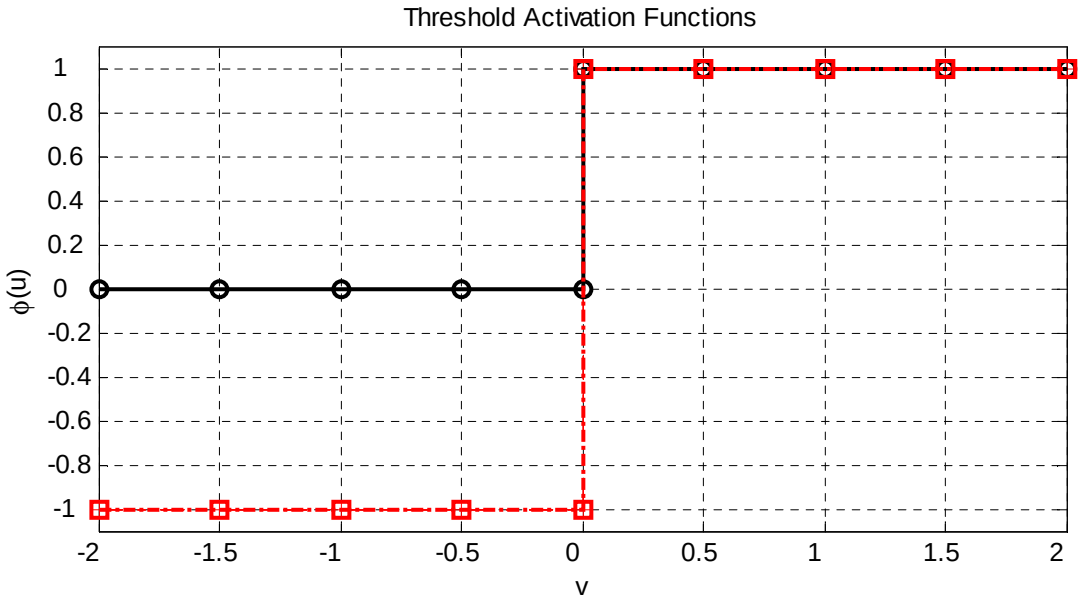


Figure 3.3 – Neuron activation functions: Threshold

Example threshold activation function equations:

$$\phi_u = \begin{cases} 1, & u \geq 0 \\ 0, & u < 0 \end{cases} \tag{3.5}$$

or

$$\phi_u = \begin{cases} 1, & u \geq 0 \\ -1, & u < 0 \end{cases} \tag{3.6}$$

Figure 3.4 shows linear activation functions where the output of the neuron is proportional to the input according to a factor *k* as described by equation (3.7), and Figure 3.5 shows a sigmoid activation function where the output is squashed between 0 and 1 with the squashing rate dependant on a factor *λ* as described by equation (3.8).

Linear activation function equation

$$\phi_u = ku \tag{3.7}$$

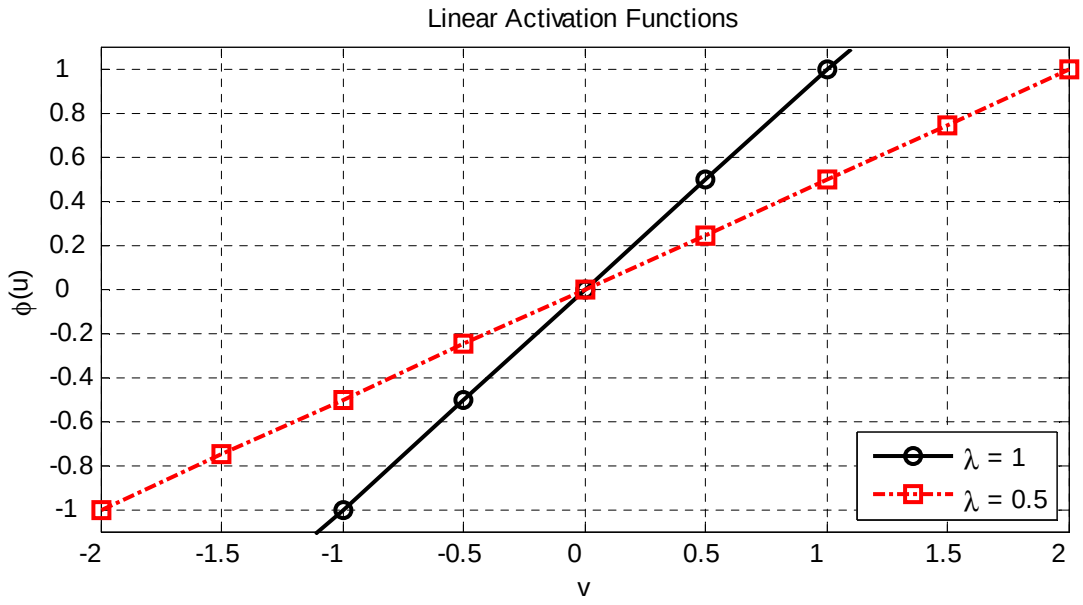


Figure 3.4 – Neuron activation functions: Linear

An alternative squashing function frequently used for neuron activation is the *tanh* function shown in equation (3.9)

Example squashing function equations:

$$\phi_u = \frac{1}{1+e^{-\lambda u}} \tag{3.8}$$

or

$$\phi_u = \tanh(u) \tag{3.9}$$

Other activation functions can also be utilised, including Radial Basis Functions (RBF) and statistical functions. Throughout this thesis, sigmoid functions are used for activation within ANN hidden layers, and linear functions are used for activation within output layers.

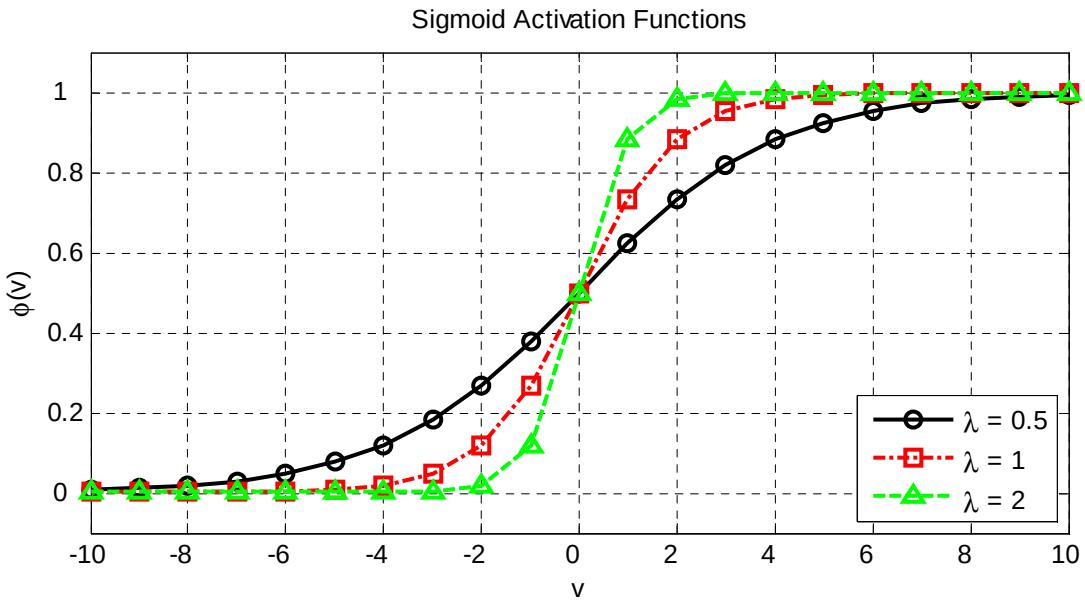


Figure 3.5 – Neuron activation functions: Sigmoid

The sigmoid function also has a straight forward analytical derivative shown in equation (3.10) which is a very useful property within many learning algorithms, i.e.:

$$\phi_u = \frac{1}{1+e^{-\lambda u}}$$

Derivative:
$$\frac{d\phi_u}{du} = \lambda\left(\frac{1}{1+e^{-\lambda u}}\right)\left(\frac{-e^{-\lambda u}}{1+e^{-\lambda u}}\right)$$

or:
$$= \lambda\phi_u(1-\phi_u)$$

or:
$$= \lambda u_k(1-u_k) \qquad (3.10)$$

3.4 ANN Architectures

Multiple instances of the basic neuron model described in Section 3.2 can be combined in very many ways to construct a wide variety of ANN architectures. A feedforward architecture of a network with one hidden layer is shown in Figure 3.6, with *n* inputs, *m* hidden neurons and *h* output

neurons. In this depiction of the network, the weights between neurons are denoted by the arrows, the summing junction and activation function are depicted as a single unit.

The input and output layers are visible to functions external to the network, while layers between the inputs and outputs are not visible and are hence 'hidden'. In this case the hidden layer uses non-linear sigmoid activation functions, and the output layer uses linear activation functions.

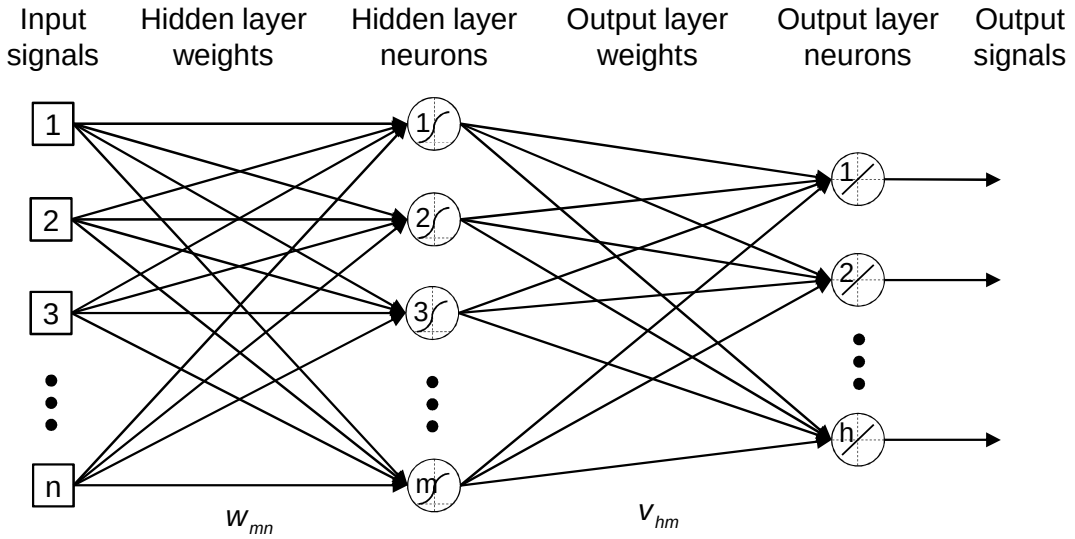


Figure 3.6 – Fully connected feed forward network with one hidden layer

The various parameters through the network can be represented as vectors and matrices:

- The inputs x are a vector of values such that $x \in \mathbb{R}^{n\times 1}$
- The weights from the inputs to the hidden layer are a matrix of values w such that $w \in \mathbb{R}^{m\times n}$
- The output of the hidden neurons via the sigmoid activation functions are a vector of values $\phi(\bullet)$ such that $\phi(\bullet) \in \mathbb{R}^{m\times 1}$
- The weights from the hidden layer to the output layer are a matrix of values v such that $v \in \mathbb{R}^{h\times m}$
- The values after the linear activation function in the output layer are a vector of values y such that $y \in \mathbb{R}^{h\times 1}$

The output values can be calculated through the network according to the matrix equation (3.11), the specific case where the output layer activation function is linear with a gradient of +1.

$$y = v \bullet \left[\phi(\mathrm{w} \bullet \mathrm{x}) \right] \tag{3.11}$$

Or, for any output layer activation function φ according to equation (3.12)

$$y = \varphi\left(v \bullet \left[\phi(\mathrm{w} \bullet \mathrm{x}) \right] \right) \tag{3.12}$$

The multiple pathways through the network, combined with the non-linear activation functions provide a very powerful capability for mapping inputs to outputs and fitting to an arbitrary function. The challenge is to find the values for the weight sets that provide a correct and robust mapping of the inputs to the outputs – not only for the datasets where input and output vectors are available, but then to predict useful outputs from input sets that have not been a part of the weight development procedure. The choice of an appropriate network architecture, and achieving training of the weights in a reliable and efficient manner presents the crux of developing an ANN application. For a given architecture, the network's knowledge is contained within the connection weight matrices.

Training methods for ANNs fall into two major categories; supervised training and unsupervised training.

Supervised training uses a set of data where the target outputs are known for the training inputs. Various paradigms for this training or learning process are used, including error–correction learning, stochastic learning and reinforcement learning. The training method is intended to find a set of network weights that minimise the error across the training dataset – a supervised learning method is able to this by understanding the gradient of the error surface, and progressing towards a minimum – though this may be a local rather than global minimum.

Unsupervised training does not use known outputs for the training input sets, and uses only local information to update weights. This can also be referred

to as self-organisation, and examples include competitive learning and Hebbian learning. These methods need some task independent measure of performance and the network is optimised on that measure, detecting the collective properties of the inputs sets presented during training.

All ANN training methods presented within this thesis use supervised learning approaches.

3.5 Feedforward Network Training – Back-propagation

One of the most utilised methods of training feedforward networks is the back-propagation algorithm. The back-propagation name is a reduction of 'backward propagation of errors' and this supervised learning method uses the error of an ANN's output compared to the target value (training value) to modify the network's weights such that the error is reduced, and over a full set of training data, the error is minimised.

The method has two phases, propagation and weight update. Propagation is the forward pass through the network followed by back-propagation of the error values (hence the method's name). Once the contribution of each weight value to the error has been found, the weights are updated in proportion.

Figure 3.7 shows the flow from an input vector through the hidden layer weights, the hidden layer activation function, the output layer weights and the output layer activation function. A training dataset will consist of many input vectors. One presentation of the full set of T input vectors is referred to as an epoch. For the forward propagation, the t^{th} input vector is weighted according to the initial hidden layer(s) weight values w (typically randomised values initially). At the j^{th} hidden neuron, the weighted connections are summed to form the induced local field U_j which is then applied to the activation function of the hidden neuron. The resulting output values Y from the hidden layer neurons are then weighted by the output layer weight values v before summing to the induced local field of the output layer neurons - for output neuron k this is u_k and is the input to the output neuron's activation

function. The forward propagation for the t^{th} training dataset produces a vector of network outputs *y(t)* according to equation (3.13) that are then compared to the corresponding target values *d(t)* and a prediction error is calculated.

$$y = \varphi(v \bullet [\phi(w \bullet x)]) \tag{3.13}$$

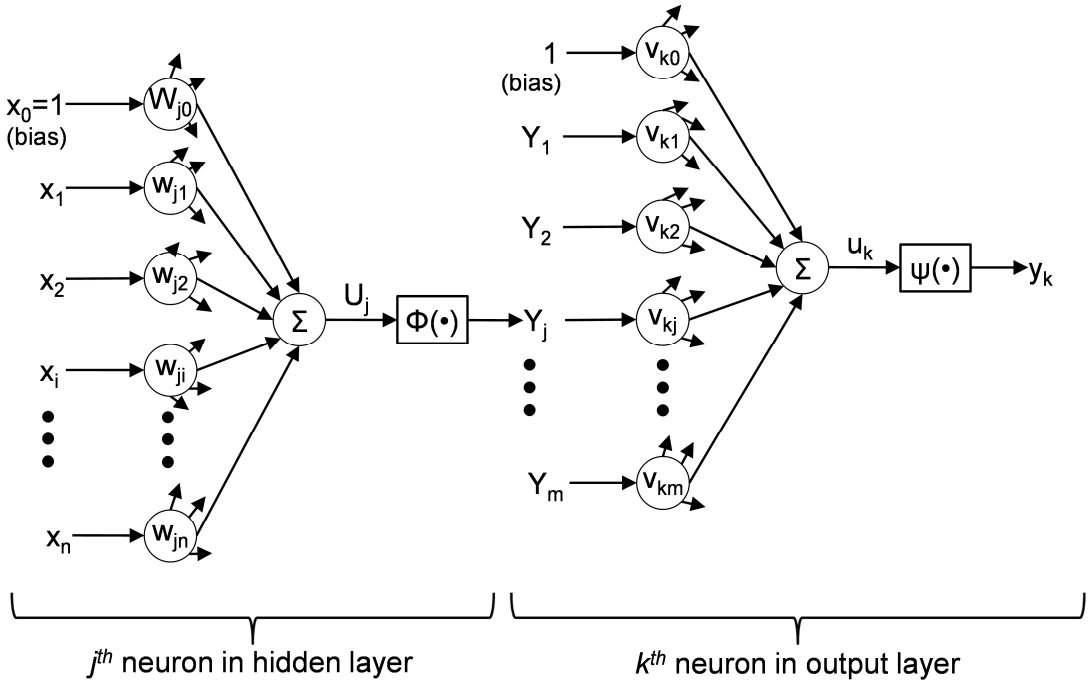


Figure 3.7 – ANN model showing j^{th} neuron in hidden layer, and k^{th} neuron in output layer

The network has *n* inputs, *m* hidden neurons and *h* output neurons:

- *x* is the input vector, with *n* elements
- *w* is the hidden layer weight matrix with *m* rows and *n* columns
- U_j is the summation of inputs to the j^{th} hidden neurons; the locally induced field : $U_j = w_{j,:} \bullet x$
- *Φ(•)* is the hidden layer neuron activation function
- Y_j is the output of the j^{th} hidden neuron's activation function acting on the locally induced field U_j: $Y_j = \phi(U_j)$
- The vector *Y* becomes the input vector to the output layer, with *m* values

- v is the output layer weight matrix with h rows and m columns
- u_k is the summation of inputs to the k^{th} output neurons, the locally induced field: $u_k = v_{k,:} \bullet Y$
- $\varphi(\bullet)$ is the output layer activation function
- y_k is the output of the k^{th} output neuron's activation function acting on the locally induced field u_k: $y_k = \varphi(u_k)$

For the t^{th} input vector, the error at the k^{th} neuron in the output layer is:

$$e_k(t) = d_k(t) - y_k(t) \tag{3.14}$$

Where $d_k(t)$ is the target output of the k^{th} output neuron for the t^{th} dataset.

The error energy cost function across the h neurons in the output layer is defined as:

$$\varepsilon(t) = \frac{1}{2}\sum_{k=1}^{h} e_k^2(t) \tag{3.15}$$

The cost function is calculated for each input vector in the training dataset. The dataset will consist of many input vectors, and if $e_k(t)$ is the error for the k^{th} neuron at the t^{th} vector presented to the network, then the instantaneous error energy for all output neurons is $\varepsilon(t)$, and the average squared error energy for the entire training dataset is:

$$\varepsilon_{av} = \frac{1}{T}\sum_{t=1}^{T}\varepsilon(t) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{2}\sum_{k=1}^{h} e_k^2(t) \tag{3.16}$$

The objective of the training process is to minimise ε_{av} by finding an optimum set of weights. The weights updates are calculated for all input vectors until one epoch (one complete set of input vectors) has been presented to the network. The arithmetic average of all the weight changes over the epoch is an estimate of the change that would result from minimising ε_{av} for the entire training dataset.

The back-propagation phase reverses the path through the network to understand how the error is influenced by each weight value. For the output weight matrix v, the first step is to understand how quickly the total error $\varepsilon(t)$ changes as the error of individual neurons changes.

This is the partial derivative of equation (3.16) with respect to $e_k(t)$:

$$\frac{\partial \varepsilon(t)}{\partial e_k(t)} = e_k(t) \tag{3.17}$$

Next, the rate at which the error of the k^{th} neuron changes according to a change in its output is the partial derivative of equation (3.14) with respect to $y_j(n)$

$$\frac{\partial e_j(t)}{\partial y_j(t)} = -1 \tag{3.18}$$

The rate at which the output of the k^{th} neuron changes with a change of its induced local field u_k depends on the derivative of the output layer activation function φ:

$$\frac{\partial y_k(t)}{\partial u_k(t)} = \varphi'(\mathrm{u}_k(t)) \tag{3.19}$$

For a linear activation function with gradient +1, this is unity, for the hidden layer with a sigmoid activation function, this is $\lambda U_j(1-U_j)$.

Finally, the rate of change of induced field u_k with change in any individual weight value is the derivative of equation (3.3):

$$\frac{\partial u_k(t)}{\partial v_{k,j}(t)} = y_k(t) \tag{3.20}$$

The rate of change of overall instantaneous error with respect to any given weight value may now be expressed in terms of the derivatives defined in equations (3.17) to (3.20) via the chain rule as shown in equation (3.21):

$$\frac{\partial \varepsilon(t)}{\partial v_{k,j}(t)} = \frac{\partial \varepsilon(t)}{\partial e_k(t)} \bullet \frac{\partial e_k(t)}{\partial y_k(t)} \bullet \frac{\partial y_k(t)}{\partial u_k(t)} \bullet \frac{\partial u_k(t)}{\partial v_{k,j}(t)}$$

$$\frac{\partial \varepsilon(t)}{\partial v_{k,j}(t)} = -e_j(t) \bullet \varphi'(\mathrm{u}_k(t)) \bullet \mathrm{y}_k(t) \tag{3.21}$$

A correction $\Delta v_{j,k}(\mathrm{t})$ is now to be applied to $v_{j,k}(\mathrm{t})$, defined by the delta rule:

$$\Delta v_{k,j}(t) = -\eta \frac{\partial \varepsilon(t)}{\partial v_{k,j}(t)} \tag{3.22}$$

where η is the learning rate parameter. The negative value forces gradient descent in weight space – a change in the weight values that results in a reduction of $\varepsilon(t)$.

Substituting equation (3.21) into equation (3.22) we have:

$$\Delta v_{k,j}(t) = \eta \bullet e_k(t) \bullet \varphi'(u_k(t)) \bullet y_k(t) \qquad (3.23)$$

The local gradient $\partial_k(t)$ is defined to be:

$$\partial_k(t) = -\frac{\partial \varepsilon(t)}{\partial u_k(t)} \qquad (3.24)$$

or, via the chain rule:

$$\partial_k(t) = -\frac{\partial \varepsilon(t)}{\partial e_k(t)} \bullet \frac{\partial e_k(t)}{\partial y_k(t)} \bullet \frac{\partial y_k(t)}{\partial u_k(t)} \qquad (3.25)$$

$$= e_k(t) \bullet \varphi'(u_k(t)) \qquad (3.26)$$

And substituting equation (3.26) in equation *(3.23)* we get:

$$\Delta v_{k,j}(t) = \eta \bullet \partial_k(t) \bullet y_k(t) \qquad (3.27)$$

Weight correction at a hidden layer

The above process derives the weight change calculation for a neuron in the output layer. The process is similar for hidden neurons, but while output neurons have direct target values to calculate their training error against, there are no immediately available target values for hidden layer outputs. This is the credit assignment problem, and is solved by back-propagating the errors through the network to each layer of neurons.

Considering Figure 3.7 once more, we can define the local gradient $\partial_j(t)$ for a hidden neuron similarly as for the output neuron definition seen in equation (3.24) as:

$$\partial_j(t) = -\frac{\partial \varepsilon(t)}{\partial U_j(t)} \qquad (3.28)$$

and using the chain rule, expand this to

$$\begin{aligned}\partial_j(t) &= -\frac{\partial \varepsilon(t)}{\partial Y_j(t)} \bullet \frac{\partial Y_j(t)}{\partial U_j(t)} \\ &= -\frac{\partial \varepsilon(t)}{\partial Y_j(t)} \bullet \phi'\left(U_j(t)\right)\end{aligned} \tag{3.29}$$

Equation (3.15) defined $\varepsilon(t)$:

$$\varepsilon(t) = \frac{1}{2}\sum_{k=1}^{h} e_k^2(t)$$

and taking the partial derivative of $\varepsilon(t)$ with respect to $Y_j(t)$:

$$\frac{\partial \varepsilon(t)}{\partial Y_j(t)} = \sum e_k \frac{\partial e_k(t)}{\partial Y_j(t)} \tag{3.30}$$

Then expanding equation (3.30) with the chain rule:

$$\frac{\partial \varepsilon(t)}{\partial Y_j(t)} = \sum e_k(t) \frac{\partial e_k(t)}{\partial u_k(t)} \bullet \frac{\partial u_k(t)}{\partial Y_j(t)} \tag{3.31}$$

Replacing $y_k(t)$ with $\varphi\left(u_k(t)\right)$ in equation (3.14), and differentiating w.r.t. u_k:

$$\begin{aligned} e_k(t) &= d_k(t) - \varphi\left(u_k(t)\right) \\ \frac{\partial e_k(t)}{\partial u_k(t)} &= -\varphi'\left(u_k(t)\right)\end{aligned} \tag{3.32}$$

The induced local field u_k at output neuron k is

$$u_k(t) = \sum_{j=0}^{m} v_{k,j}(t) \bullet Y_j(t) \tag{3.33}$$

And differentiating w.r.t. $Y_j(t)$ gives:

$$\frac{\partial u_k(t)}{\partial Y_j(t)} = v_{k,j}(t) \tag{3.34}$$

The derivative $\frac{\partial \varepsilon(t)}{\partial y_j(t)}$ can now be constructed using equations (3.31), (3.32) and (3.34):

$$\frac{\partial \varepsilon(t)}{\partial Y_j(t)} = -\sum_{k=1}^{h} e_k(t) \bullet \varphi'\left(u_k(t)\right) \bullet v_{k,j}(t) \tag{3.35}$$

And substituting $\partial_k(t)$ for $e_k(t) \bullet \varphi'(u_k(t))$ from equation (3.26):

$$\frac{\partial \varepsilon(t)}{\partial Y_j(t)} = -\sum_{k=1}^{h} \partial_k(t) \bullet v_{k,j}(t) \tag{3.36}$$

The back-propagation formula for the hidden layer local gradient ∂_j is found by using equation (3.36) in equation (3.29):

$$\partial_j(t) = \phi'(U_j(t)) \bullet \sum_{k=1}^{h} \partial_k(t) \bullet v_{k,j}(t) \tag{3.37}$$

And the hidden weight update is performed according to:

$$\Delta w_{j,i}(t) = \eta \bullet \partial_j(t) \bullet Y_j(t) \tag{3.38}$$

In summary, the steps for back-propagation method are:

- present an epoch of training input vectors to the network, with the hidden and output weight matrices initialised to random values
- for each input vector, propagate through the network in the forward direction to find the vector of output values, compare these to the target values and calculate the error for each output neuron
- propagate the errors backwards through the network, calculating the local gradients at each neuron
- adjust the weight value on each network connection according the delta rule, dependant on the local gradient
- re-present the epoch of training data and repeat the training steps until a stopping criteria is reached – usually a limit on the change of values between epochs

Pre and Post Processing of Data

Typically input vectors and target vectors are pre-processed prior to use in the training exercise. The values of engineering units may vary wildly and require a very wide dynamic range that would give the network an additional task to accommodate. An example is presenting cylinder pressure in the SI unit of Pascals results in a large number – typically in the range 2×10^6 – 5×10^6, while other signals from the engine measurements transducers may be numerically small – the knock sensor lies in the range ±10v. To avoid this

very large range of values, all signals are first scaled to an approximate range of ±1 prior to use in training of neural networks. The same scaling factors used for each signal for training are also applied to the signals input to a network when testing their predictive capabilities on unseen data. Network predictions, which will normally be in the range of approximately ±1 due to the initial scaling of target values and the squashing influence of the sigmoid function are generally rescaled back to engineering units using the same factor used to pre-process the signals prior to presentation to the network.

3.6 Recurrent networks

Recurrent Neural Networks (RNNs) are characterised by feedback connections within the network. These feedback connections can take a wide variety of forms. Feedback paths may be local, with layer outputs feeding back to the input of the same layer, one or more steps delayed (a feedback cannot occur until the relevant value in the network has been calculated on the previous propagation). Alternatively, feedback paths may be global, with values calculated at the output layer of the network on the previous propagation feeding back to the input layer for the next step.

Such a global feedback network is presented in Figure 3.8 with the outputs *y(t)* calculated on the preceding step, fed back to be inputs on the next step. Such a network structure is known as Non-linear Autoregressive with Exogenous inputs (NARX).

The Non-linear title acknowledges the behaviour of the multi-layer perceptron that forms the core of the network, the Autoregressive term acknowledges the fact that the network feeds back its own outputs, and the Exogenous inputs refer to the input vector *x(t)* that is independent of the feedback loops. In addition, both the feedback terms, and the exogenous input vector are shown with additional time delay terms, combining recurrent and time delay architectures. Such recurrent, time-delayed network architectures, introduce

temporal features to the processing, with memory and a state representation capability.

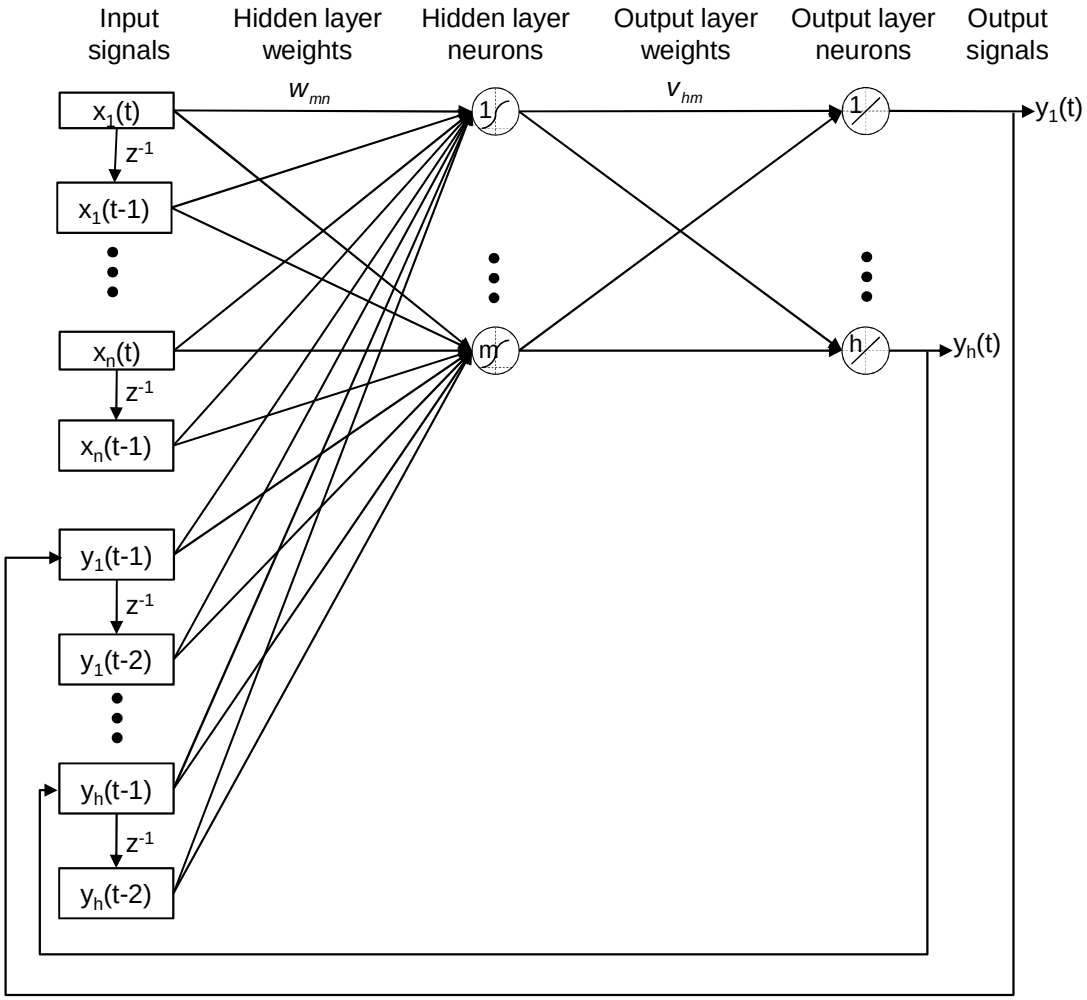


Figure 3.8 – Recurrent neural network with NARX structure

RNNs offer great potential processing capability. [Haykin, 1999] summarises the capabilities with two quoted theorems:

i) ‘All Turing machines may be simulated by fully connected recurrent networks built on neurons with sigmoid activation functions.’ [Siegelmann and Sontag, 1991]

ii) ‘NARX networks with one layer of hidden layer neurons with bounded, one-sided saturated activation functions and a linear output neuron can simulate fully connected recurrent networks with bounded, one-sided saturated functions, except for linear slowdown.’ [Siegelmann et al., 1997]

So, while the NARX architecture is relatively simple compared to that possible with more hidden layers, and with more complex recurrent connections, the capability can be similar if the network can be successfully trained.

However, the recurrent features introduce a significant challenge to network training. Although the forward propagation for any of the input vector set through the network functions in just the same way as the feedforward architecture described above, the next input vector cannot be constructed until the previous prediction is complete and is now dependent on the weight matrices of the network which are changing during training. Various paradigms for NARX network training are available, among them Standard Back-Propagation using Teacher Forcing (SBPTF), Back-Propagation Through Time (BPTT), Real Time Recurrent Learning (RTRL), and the use of Kalman Filtering to improve the use of the available data.

Chapter Four

4. A DESCRIPTION OF THE ENGINE TEST FACILITIES AND DEVELOPMENT OF A DATA ACQUISITION SYSTEM FOR OBTAINING SYNCHRONOUS MEASUREMENTS

High quality engine test data is seen as a pre-requisite for input to the neural network training activity that forms the core of this thesis. This chapter first details the engine used to generate the data and the test facility in which it is installed. Next, the instrumentation available on the engine is described, and the data acquisition hardware and software used to digitise the signals are explained. Finally, the signal processing methods employed to reduce the data to make it suitable for application to ANN training are covered.

Of particular interest are the approaches to overcome inaccuracies in the crankshaft angular encoder used to generate crank kinematic data, the methods of synchronising time based and angular based data acquisition, the alternative crank kinematic processing from an inductive probe targeted on the flywheel teeth and the extraction of time-frequency data from the cylinder block acceleration signals.

4.1 Ford I3 Engine

The engine used to generate the operating data for input to the ANN training effort was a gasoline fuelled inline 3-cylinder (I-3) configuration engine supplied to University of Sussex by the Ford Motor Company. Designed in collaboration with Yamaha, the engine is a prototype build and the model was not put into production.

The engine is an aluminium (head and cylinder block) 3-cylinder 4-stroke unit with a 120° crank throw. The combustion system is direct injection, spark ignited (DISI). The cylinder head is a 4 valve per cylinder design (2 intake, 2 exhaust) with swirl control valves on the inlet and an exhaust gas recirculation (EGR) system. The valve train is belt driven, and the crank nose carries a torsional vibration damper. The flywheel is connected to the

dyno driveshaft via a torsionally compliant coupling, no clutch elements are installed.

Viewed from the rear (flywheel) end of the engine, the exhaust system is on the right, and the intake system on the left. Throughout this thesis the conventional cylinder numbering system is used, with cylinder 1 being furthest from the flywheel, and cylinder 3 closest to the flywheel.

Table 4.1 gives the primary engine parameters of interest for the current work. Figure 4.1 shows the engine installed to the gasoline research test bed at University of Sussex.

Engine Kinematic Parameters	Value
No. of cylinders	3 Inline
Bore	79.0 mm
Stroke	76.5 mm
Swept Volume	1125 cc
Connecting rod length	137 mm
Piston pin offset	0.8 mm
Compression ratio	11.5
Piston mass	270 gram
Connecting rod mass	395 gram
Crankshaft primary inertia	0.02579 kgm^2
Flywheel inertia (including driveshaft and adapter)	0.12021 kgm^2
Number of flywheel teeth	135

Table 4.1 – Specification of Ford I-3 test engine

The 3 cylinder engine configuration is considered ideal for this work. The low number of cylinders results in wide spacing of the combustion events though the crank rotation cycle, helping to avoid temporal interactions between the combustion events and other events within the engine (piston slap, valve opening and closing etc.). However, the 3 cylinder format is also highly applicable to future production engines as gasoline engine downsizing becomes a production reality.

4.2 Test Facility

The Ford I-3 Engine was installed in the gasoline research laboratory at University of Sussex.

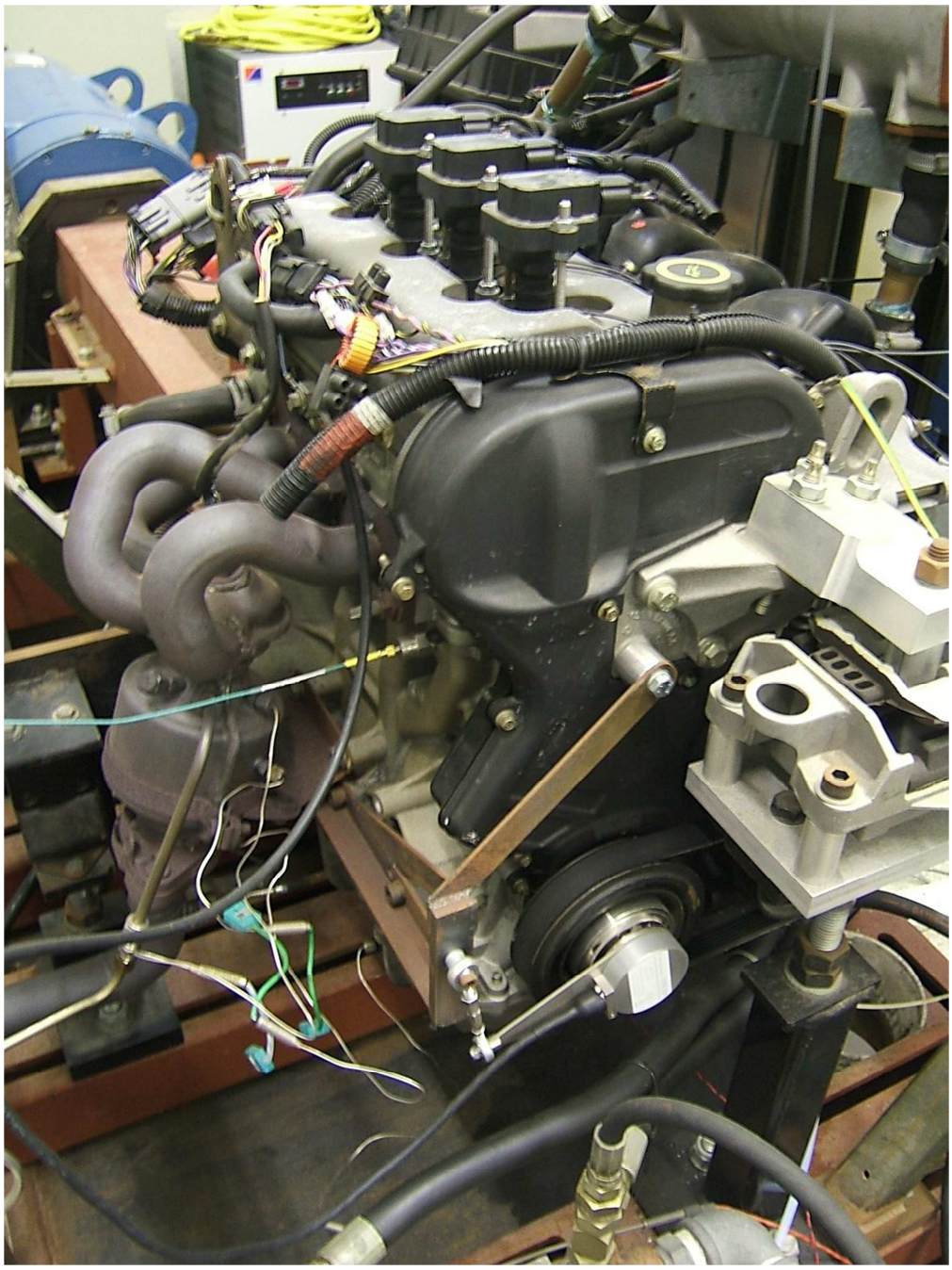


Figure 4.1 – Ford I-3 engine installed on gasoline research test bed

The engine's output power is absorbed by a McClure 130kW / 7000rev/min DC dynamometer, controlled by a Eurotherm control cabinet. Figure 4.1 shows the engine on the test bed, Table 4.2 lists the specification of dynamometer.

Dyno Parameter	Value
Maximum absorption power	130 kW
Maximum motoring power	100 kW
Maximum speed	7000 rev/min
Armature inertia	0.87 km^2
Coupling torsional stiffness	1260 Nm/rad

Table 4.2 – Specification of McClure DC dynamometer

The dyno is mounted in a rotating frame, and torque measured using a load cell on a moment arm. The engine control system is manual; with set points for either speed or torque and manual control of the engine's throttle angle. More complex control of the engine is possible via INCA calibration software access to the ECU, but this is not employed for the current work. Engine operational monitoring consists of oil and coolant temperatures, engine speed and dyno torque. Before acquiring any data, the engine was allowed to thoroughly warm up to the oil and coolant set-points, each at 90°C.

Engine / driveline / dynamometer torsional characteristics

The natural frequency of 2 inertias connected by a coupling is given by equation (4.1):

$$f_n = \frac{1}{2\pi}\sqrt{k\frac{I_1 + I_2}{I_1 I_2}} \tag{4.1}$$

Where:

f_n = Natural frequency of 2 inertias system [Hz]
k = Stiffness of coupling [Nm/rad]
I_1 = Moment of inertia, mass 1 [kgm^2]
I_2 = Moment of inertia, mass 2 [kgm^2]

Using the relevant values from Table 4.1 and Table 4.2, the first system natural frequency is found to be approximately 16.5 Hz. With the 3 cylinder engine firing at 1.5 times the rotational frequency, the critical engine speed would equate to 660 rev/min. Allowing an isolation factor of √2, the allowable region of engine operations is >905 rev/min. All test data was acquired at engine speeds of 1000 rev/min or above.

A schematic diagram of the engine installation on the test bed is shown in Figure 4.2.

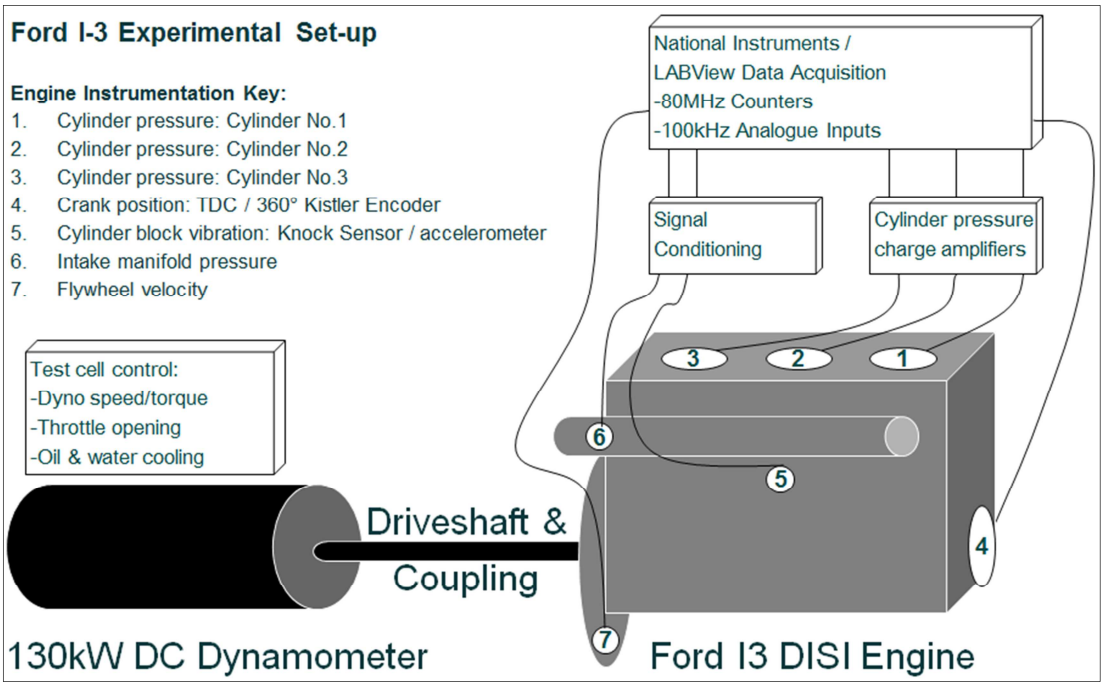


Figure 4.2 – Schematic of Ford I-3 engine test bed arrangement and National Instruments data acquisition system

4.3 Engine Instrumentation

The following instrumentation sensors were installed on the Ford I-3 engine for this work:

- Cylinder pressure for all 3 cylinders
- Crank nose 360 pulse/revolution encoder with 1/rev TDC mark
- Cylinder block vibration using an additional standard knock sensor on the intake side of the engine
- Cylinder block vibration using a piezo-electric accelerometer on the exhaust side of the engine
- Intake manifold pressure sensor
- Flywheel rotational velocity via an inductive probe targeted on the flywheel teeth

Cylinder pressure

The cylinder pressure transducers employed were Kistler type 6117BCD36 spark plug integrated transducers with a 0-150 bar range. This type of transducer is particularly convenient for research work as they require no modification to the cylinder head for installation. The specific transducers installed throughout all data acquisition are listed in Table 4.3.

Cylinder Number	Transducer Serial Number
1	1282636
2	1346612
3	1346611

Table 4.3 - Serial numbers of Kistler cylinder pressure transducers

The signals from the pressure transducers are carried via low noise charge cables to Kistler Type 5044 charge amplifiers, inputs individually set to the transducer charge sensitivities and the output gains set to 10 bar/volt.

Crank kinematics

Crank angular displacement was measured at the crank nose using a Kistler Type 2614A1 optical encoder, with signal conditioning through a Kistler Type 2614A4 pulse multiplier. This encoder offers 2 output streams:

- 1 pulse per revolution (ppr)
- 360 or 3600 pulses per revolution

The encoder was installed with a stiff bracket, rigidly braced to the cylinder block such that vibration amplitudes were minimised. The connection to the crank is torsionally very stiff, and the rotating element of the encoder has a low inertia, ensuring the motion of the encoder is that of the crank nose. The instrument was carefully aligned such that the rising edge of the 1 pulse per revolution was coincident with piston top dead centre (TDC). The 2nd output pulse trained was configured for 360 ppr (i.e. 1° crank rotation pulse spacing). The 3600 ppr option was found to be unreliable as detailed in section 5.5.2. The rising edges of the 1 ppr (at TDC) and the first 1 degree marker were coincident.

Some challenges with data from this industry standard encoder and set-up were encountered due to apparent limited accuracy of the 1° pulses. With the primary use of the crank kinematic data to derive crankshaft acceleration, the small variations in the displacement of the pulses resulted in high levels of noise after numerically differentiating twice to angular acceleration. The novel method of calibrating the encoder to minimise this issue is described in Section 5.5.1.

The optical encoder was intended to provide a 'gold standard' measurement of crankshaft kinematics. However, cost, robustness and packaging requirements of the instrument would be a significant challenge for use in production. To provide a comparison of crank kinematics derived from a production compatible solution, an additional signal was acquired from an inductive probe targeted on the flywheel teeth. Such a signal is similar to that already available from the ECU's crank position sensor, and even if a duplicate sensor was required, the cost would be low. The frequency demodulation method used to extract rotational velocity from this signal is described in section 5.6. Comparison of kinematic data from the flywheel with that from the crank nose also allowed torsionally flexible crankshaft behaviour to be studied.

Cylinder block Vibration

Cylinder block vibration was measured using 2 transducers – a standard Bosch 261-231-114 knock sensor and an instrumentation quality accelerometer. Previous work on this engine [Vulli 2006] had found that connection of the standard fitment sensor to a data acquisition system as well as to the ECU resulted in too low a signal to be useful. Hence the original sensor remained connected to the ECU while an identical sensor was attached to the engine (also on the left / inlet side) and connected to the data acquisition system via a Laplace Instruments VIP-20 amplifier.

The knock sensor is seen as an ideal production solution – most if not all gasoline engines already have knock sensors installed, and hence no additional sensor cost would be necessary to implement this measurement in production. However, location and signal quality (bandwidth, frequency

response) of the knock sensor are currently optimised for knock sensing and may not be suitable for cylinder pressure reconstruction. To mitigate this concern, an instrumentation quality accelerometer, a Sensonics PZP1 piezo-electric sensor (0-20kHz, 0-600g), was installed on the right (exhaust) side of the engine mounted in a bolt boss. The signal from this accelerometer was carried by low noise charge cable to a Kistler Type 5044 charge amplifier.

4.4 Data Acquisition

The data acquisition system is based on National Instruments (NI) hardware, controlled and programmed using LabVIEW software.

4.4.1 Hardware

The data acquisition hardware is a NI PXI system consisting:

- PXI-1031 Chassis
- PXI-8331 Interface to Windows PC
- PXI-6133 Analogue input module with 14 bit synchronous sampling across 8 channels using a TB-2709 terminal block for low noise co-axial cable connection with a maximum sample rate of 2.5MS/s, and a maximum input amplitude of 10v
- PXI-6602 Counter/timer with 32-bit counters and 80MHz maximum source frequency, using a BNC-2121 terminal block for low noise co-axial cable connections

The 8 channel 14-bit analogue input module is used for all sensor inputs other than the crankshaft encoder which is read by the counter/timer unit. The PXI-6133 offers a very high sample rate capability combined with a dynamic range typical of engine combustion analysis systems.

4.4.2 Crank-angle based- versus time-based data acquisition

The NI PXI-6133 analogue input card is capable of digitising data clocked by an external clock signal or by an internal time-base. This offers two routes to acquiring the analogue channels described as follows:

Constant-angle sampling, acquisition externally clocked from the degree-pulses generated by the Kistler encoder

This approach is normal for combustion analysis, where the data is primarily inspected and reduced in the crank angle domain. However, the data to be supplied for training of a neural network was desired to be of equi-spaced time intervals; hence data sampled in the crank angle domain will later require re-sampling to a constant time-base. A more significant problem is that constant crank-angle-clocked data has an effective sampling frequency which varies with engine speed: i.e. at 1000 rev/min, a 1° sampling interval results in a 6 kHz sample rate (6000 rev/min equates to 36 kHz). With no anti-aliasing protection available on the analogue input card, the lower rates are not sufficient to be confident that data will not be corrupted. The encoder offers 0.1° resolution which could be used to increase the sampling frequency towards an acceptable rate, but the extrapolation method generating the 10x multiple of 1° pulses appears unreliable, and consequently generates very noisy crank kinematic data.

Constant time-base sampling (equal time intervals between samples), clocked by the PXI-6133 internal time-base

Using the analogue input card's internal time-base, samples can be clocked at up to 2.5 MHz, and a data rate can be determined which avoids most aliasing issues. The main issue then becomes one of ensuring synchronisation between the analogue signals and the crank kinematic signals. This is achieved by using the TDC pulse from the encoder as a common trigger to initiate acquisition on all inputs simultaneously. This approach starts both acquisitions within nanoseconds of each other, and thereafter, any drift between the signals will be controlled by the accuracy of the analogue time-base and the counter resolution – there is no detectable drift over the duration required for the capture of say, 200 engine cycles at 1000 rev/min.

Primarily as a consequence of aliasing concerns, and the realisation that the 0.1° encoder pulse train generates unacceptable crank kinematic data (which cannot be corrected) constant time-base sampling has been chosen as the

most robust data acquisition approach. The data acquisition/reduction process is shown graphically in Figure 4.3 and Figure 4.10.

4.4.3 Methodology for synchronisation of analogue signal and crank kinematics

The measurement signals fed to the National Instruments (NI) data acquisition system are schematically described in Figure 4.3.

All analogue voltages are synchronously sampled at the PXI-6133 with a fixed sample frequency. All analogue input channels are streamed to a single file in the NI native TDMS (Technical Data Management Streaming) format, a binary file format. This file is represented by Ana.TDMS in Figure 4.3. Figure 4.4 shows the LabVIEW block diagram defining the acquisition control.

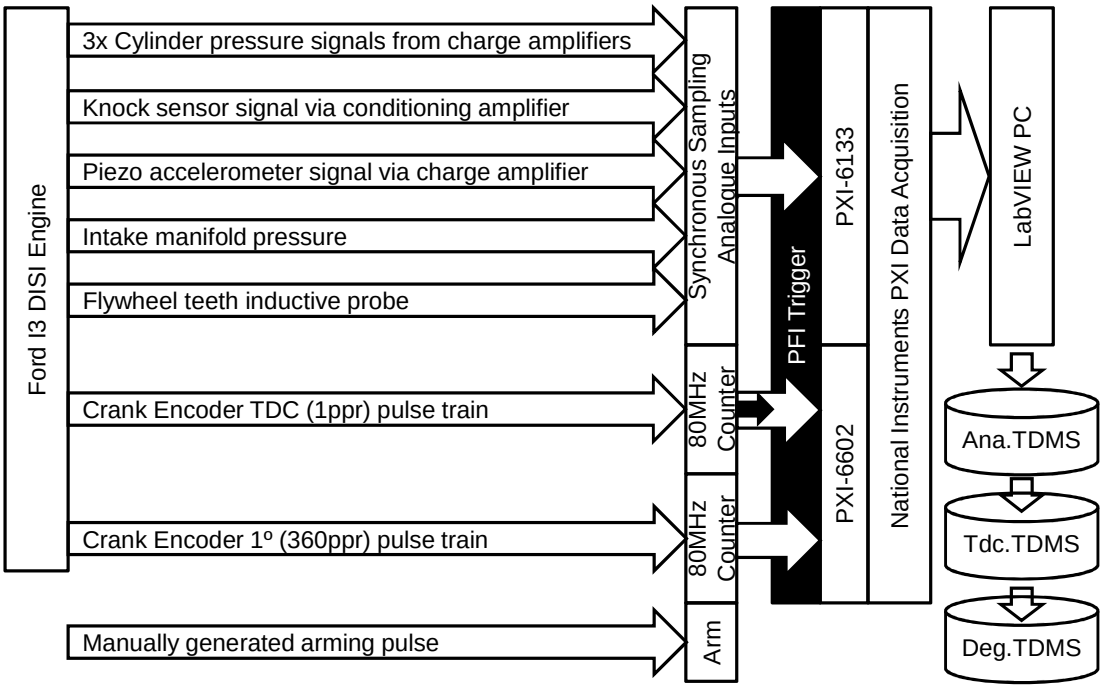


Figure 4.3 –Schematic of data signal connection to NI data acquisition system

The TTL pulse trains generated by both the 1 ppr and the 360 ppr encoder outputs are input to separate 80 MHz timers at the PXI-6602. The time intervals between each rising edge of the pulse trains are written to separate TDMS files – i.e. Tdc.TDMS for the 1ppr channel, and Deg.TDMS for the 360 ppr channel.

The need for separate files is driven by the different methods of sampling. The data from counters and from fixed time interval sampling regimes are not directly comparable – there is no fixed relationship between the times at which the two data types are taken. The different data streams and resulting multiple files require some care to ensure they are synchronised correctly. All data streams are initiated at the rising edge of an arbitrary TDC pulse from the 1ppr encoder channel, this pulse being shared on the hardware trigger PFI lines intended specifically to achieve high time synchronisation across the various PXI modules.

However, early tests using software arming of the acquisition system showed that the different data streams through the PXI system could become ready and arm at slightly different times, and would then trigger on the first available TDC rising edge to reach them. If another data stream was not armed and ready to initiate at the same TDC point, then the data streams could become out of step by some integer number of TDC pulses. Recognising how many crankshaft rotations had occurred between the first and subsequent data streams initiating proved unreliable, and an additional step was devised to control the arming of the data streams and therefore to enforce the desired simultaneous start of all data streams.

To this end, a 4^{th} input type to the data acquisition system is used, this being a manually generated hardware TTL arming pulse which is synchronously available across all hardware boards via another PFI line. The BNC-2121 terminal block provides a button controlled TTL generation capability that was utilised for this purpose. The LabVIEW code was modified such that each data stream would not be available to initiate acquisition on a TDC pulse until it had received the arming pulse. Figure 4.4 shows the LabVIEW block diagram defining the acquisition control.

Now, on starting the software, all background preparation actions could be completed during a short pause of 5-10 seconds, and each of the data stream allowed to become ready to acquire.

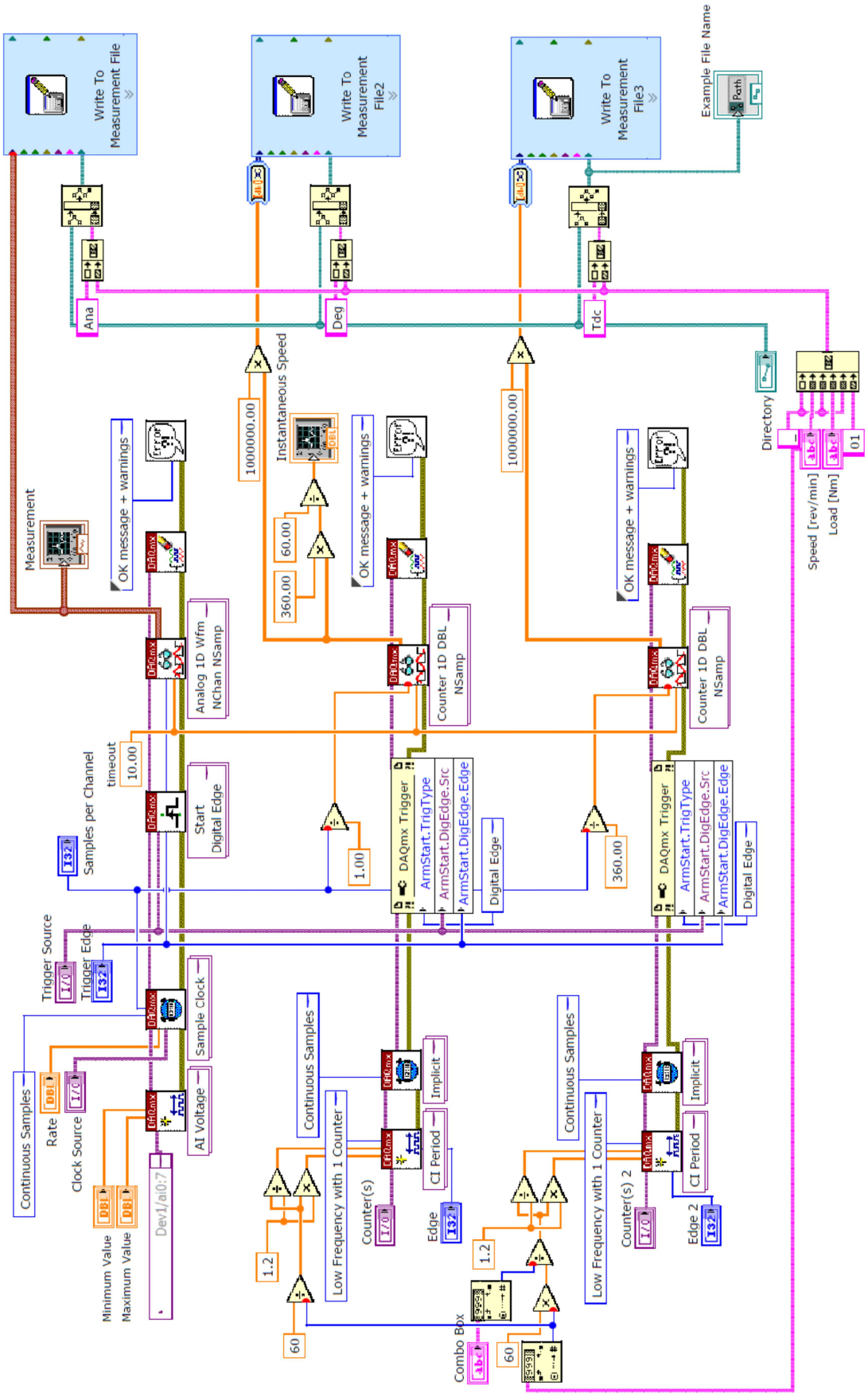


Figure 4.4 – Block diagram of LabVIEW data acquisition programme

They would then hold until armed by the manually generated pulse, and all would then initiate acquisition simultaneously on the next TDC pulse rising edge. The zero time reference for the 3 files written to disk containing the analogue inputs, the TDC counter time intervals and the degree counter time intervals would then all be synchronised. Figure 4.5 shows a schematic of the data streams, the manually generated arming pulse and the TDC trigger.

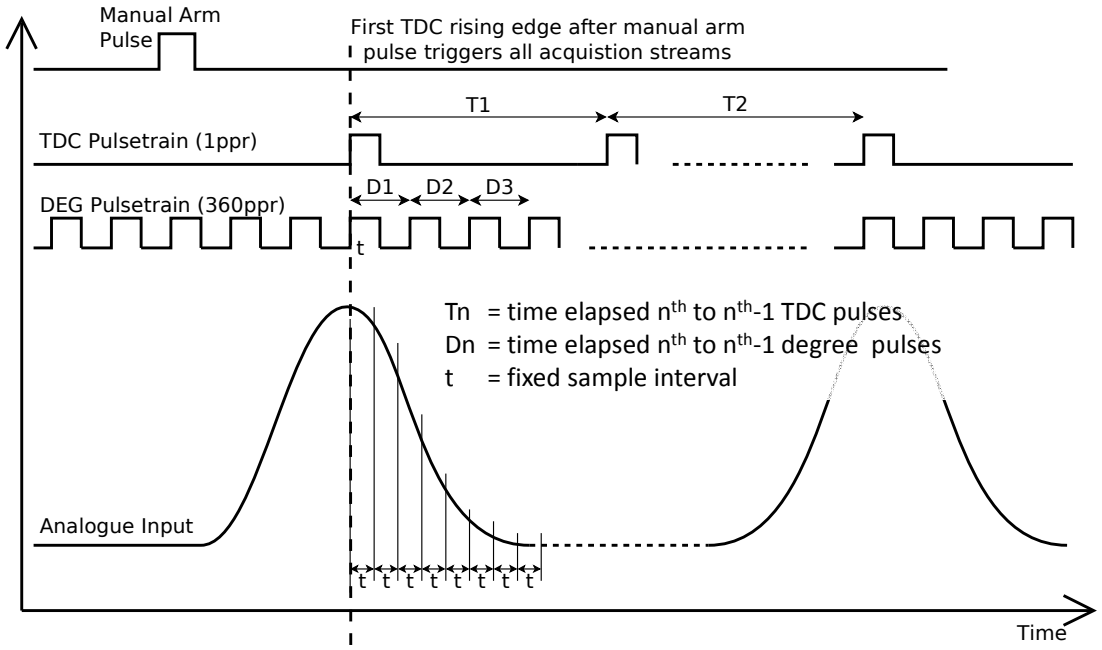


Figure 4.5 – Schematic of data streams and manual trigger pulse

4.4.4 Analogue data acquisition rates

The NI PXI-6133 module used to acquire analogue inputs is capable of digitising the signals at very high rates sample rates to a maximum of 2.5 MHz. It is important to select the sample frequency carefully, to capture information at frequencies of interest, and to avoid aliasing of data while minimising data file sizes to allow sufficient overall duration of data to be acquired and efficiently processed.

The PXI-6133 module does not incorporate analogue anti-aliasing filters, which opens the digitisation process to the risk of high frequencies (above

those of interest to the question under investigation) being folded back about the Nyquist frequency to corrupt information which is of interest. To ensure this problem was avoided, a short study was conducted to understand the frequency content of the various analogue signals being acquired.

The signal with the highest significant frequency content was seen to be the knock sensor signal, and the investigation of required sample rate was conducted on this signal.

With the engine running at 2000 rev/min, at maximum load to provide high levels of energy input to the knock sensor, a set of data was acquired from the knock sensor data channel at a sample rate of 1 MHz. This data was then post processed using the Fast Fourier Transform (FFT) to understand the wide band frequency content. The black curve in Figure 4.6 shows that the energy content at frequencies above 125 kHz is very low which suggest the use of 1 MHz as a base comparison sample frequency is valid. There are significant energy levels in the signal up to 100 kHz.

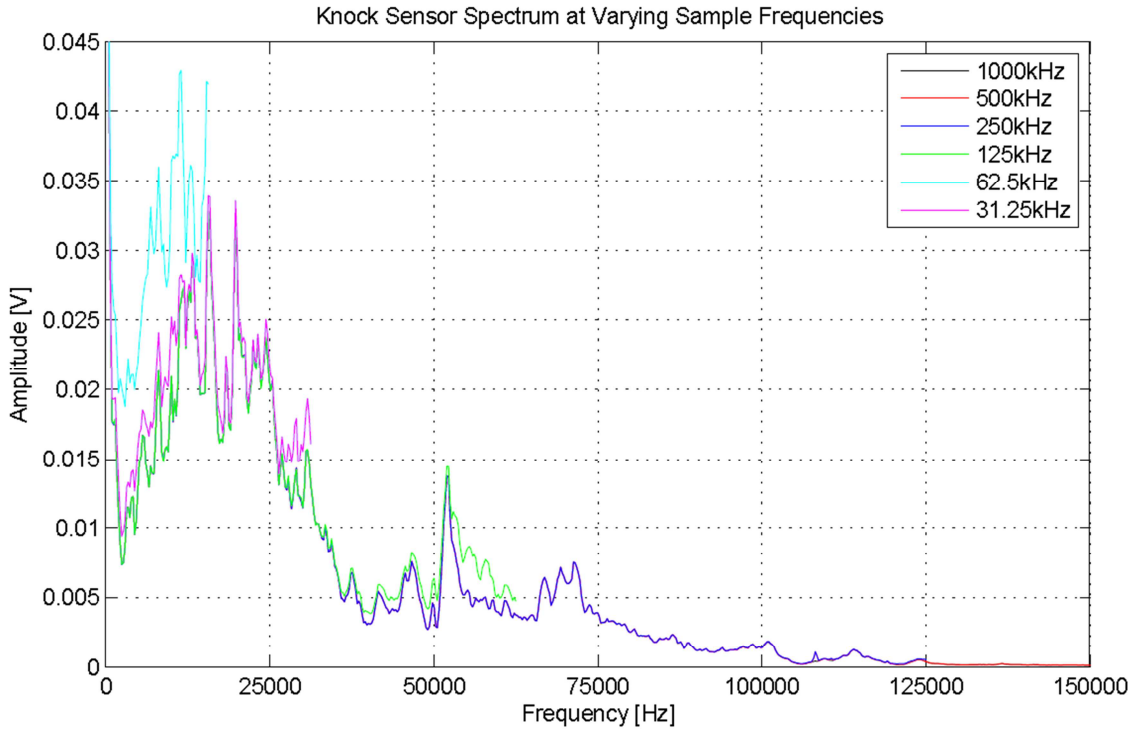


Figure 4.6 – Comparison of apparent knock sensor signal frequency content at varying sample frequencies

The 1 MHz data was then decimated by a series of factors of 2 to see the resulting effect on the apparent frequency content. Down-sampling gives a series of lower sample frequencies without alias protection. Figure 4.6 shows that down-sampling by factors 2 (equivalent sampling frequency 500 kHz) and 4 (equivalent sampling frequency 250 kHz) produce no significant change relative to the 1 MHz baseline. Further down-sampling begins to show the problem of aliasing as frequencies above the Nyquist frequency appear as corruption of lower, real frequencies. The data down-sampled by a factor of 8 (equivalent sampling frequency 125 kHz) shows a clear divergence from the baseline in frequencies around 55 kHz, and the range 25-50 kHz is affected to a lesser level. Down-sampling to equivalent sampling frequencies of 62.5 kHz and 31.3 kHz result in significant deviation from the 1MHz baseline even at frequencies below 15 kHz.

Figure 4.7 presents a comparison of the spectra derived from the different decimation rates as a ratio against the 1 MHz spectrum.

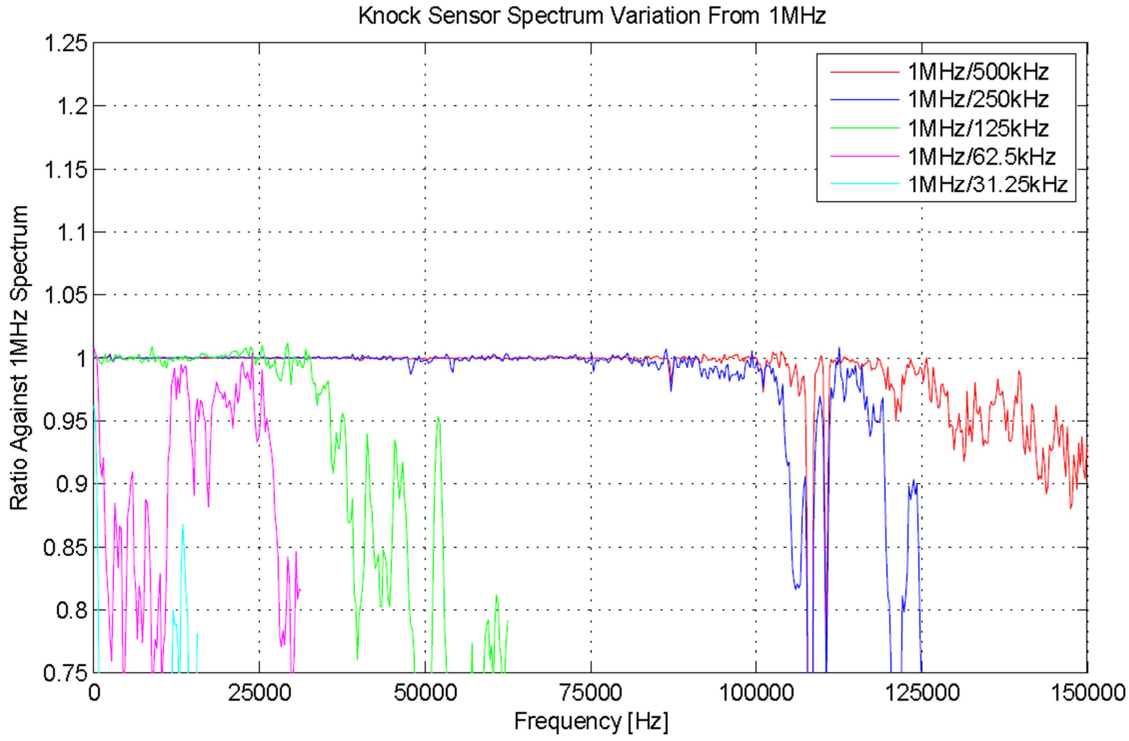


Figure 4.7 – Ratio of apparent knock sensor signal frequency content at varying sample frequencies against 1MHz baseline

For data at an equivalent sample rate of 125 kHz, the variation of energy content from the baseline is better than 5% at frequencies < 50 kHz, and better than 1% at frequencies < 25 kHz.

As the frequencies of interest from the knock sensor are anticipated to be below 20 kHz, acquiring the analogue data at a fixed sample rate of 125 kHz was concluded to be sufficient, and offered an acceptable compromise between the overall duration of data that could be recorded (i.e. file size) and alias protection of the signals. Once the signals had been digitised, further sample rate reduction in post processing involving digital anti-alias filtering and down-sampling could be readily achieved.

4.4.5 Noise Suppression

The objective of the data acquisition process was to achieve a high quality, reference dataset that could be used for ANN training. Ensuring the best possible quality of the data was important.

Early test data was found to be contaminated with occasional noise spikes, particularly from the cylinder pressure transducers and from the cylinder block mounted piezo-electric accelerometer. Piezo-electric transducers can be subject to noise problems – they generate very low levels of signal prior to amplification, and the engine and test cell are high noise environments due to high energy spark systems and the inverter drive for the dynamometer. The noise spikes were eliminated by a combination of careful routing of signal cables, and common earthing of the test bed frame, the charge amplifier casing, and the data acquisition module.

All data was checked for obvious noise issues after recording. Figure 4.8 show example data recorded prior to resolution of noise problems.

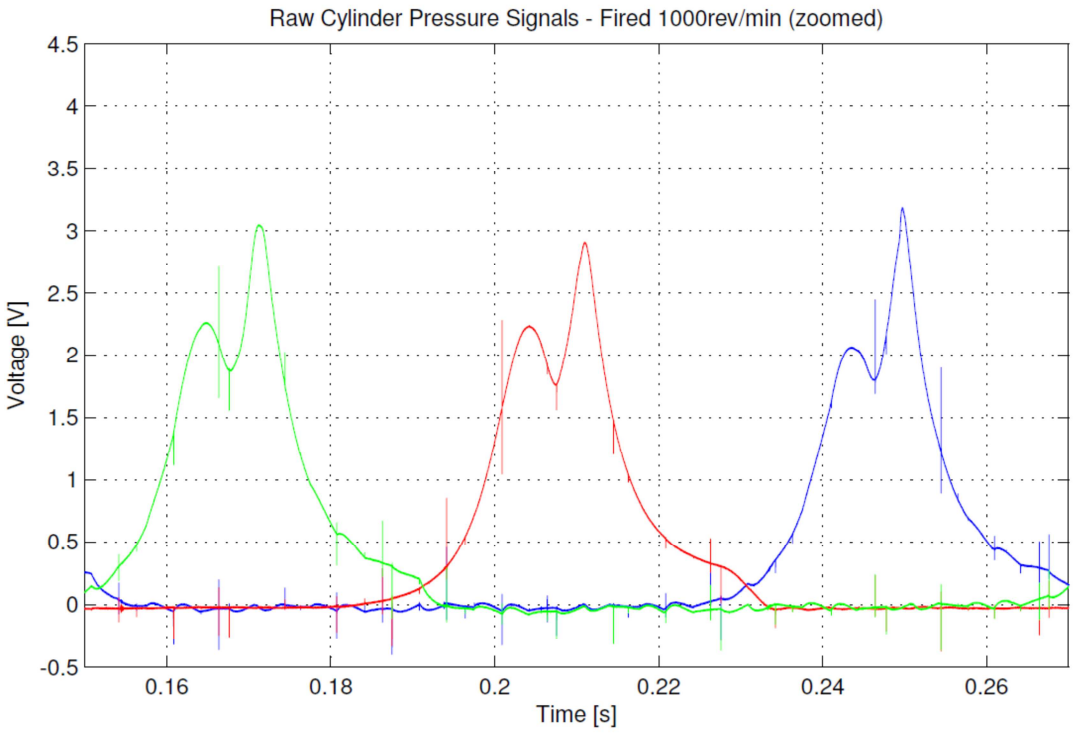


Figure 4.8 – Example cylinder pressure with noise spikes prior to revisions to cable routing and instrumentation earthing

4.5 Combination and synchronisation of encoder and analogue signals to a single dataset

As described in Section 4.4.3 and shown schematically in Figure 4.3, data from each of the 1 ppr (TDC) and 360 ppr (1°) channels of the encoder, and the analogue channels are acquired through separate streams, and written to separate TDMS files. These datasets need to be brought together into a single post processed and synchronised set to enable efficient use for ANN development.

Each of the files has a common start time, being each triggered by the first TDC pulse following manual arming. The analogue inputs are sampled at equal time steps while the encoder data consist of a series of time values representing the intervals between encoder pulses. The objective of the combination process is to bring the encoder data into the same equal time step sequence as the analogue data. Figure 4.9 shows a schematic of the data.

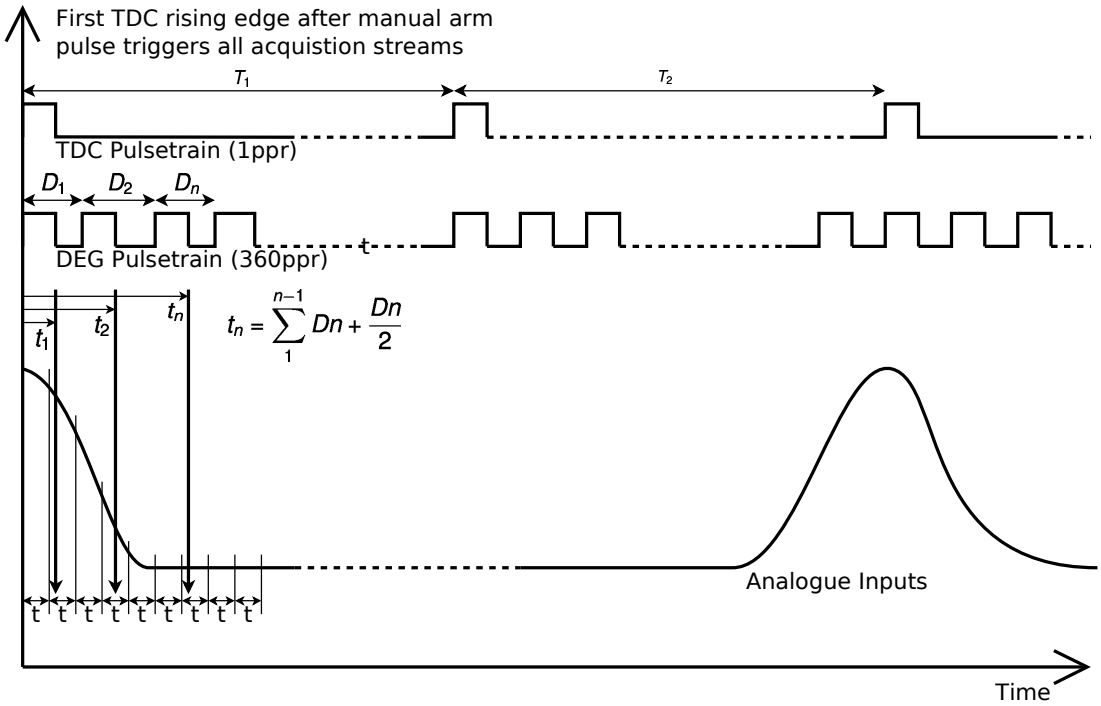


Figure 4.9 – Combination and synchronisation of encoder and analogue input data

Each 1° encoder duration is used, in conjunction with the encoder pulse angular calibration information described in detail in section 5.5.1, to generate the mean speed across that duration by dividing the duration into the calibrated angular displacement. The cumulative sum of previous degree durations is added to half the current degree duration to provide the time stamp for this 1° mean rotational velocity.

The result is a sequence of rotational velocities with non-equal spaced timings. These velocities are then resampled to the same time intervals as the analogue channels to result in a full dataset including the crank kinematic data all with equal spaced time steps. Finally, the high sampling rate of all data (driven by the need to avoid aliasing at the time of data acquisition) can be reduced by digital anti-alias filtering and down sampling. The data is reduced to 10kHz sample frequency, first and last cycles removed to eliminate filtering end effects and the start point of the data set to the time point nearest to the first TDC non-firing cylinder no. 1 in order to ensure all data presented to ANNs have the same starting behaviour. Finally the data is written to a MATLAB file for use during ANN development. Figure 4.10 shows a flow chart of the data combination.

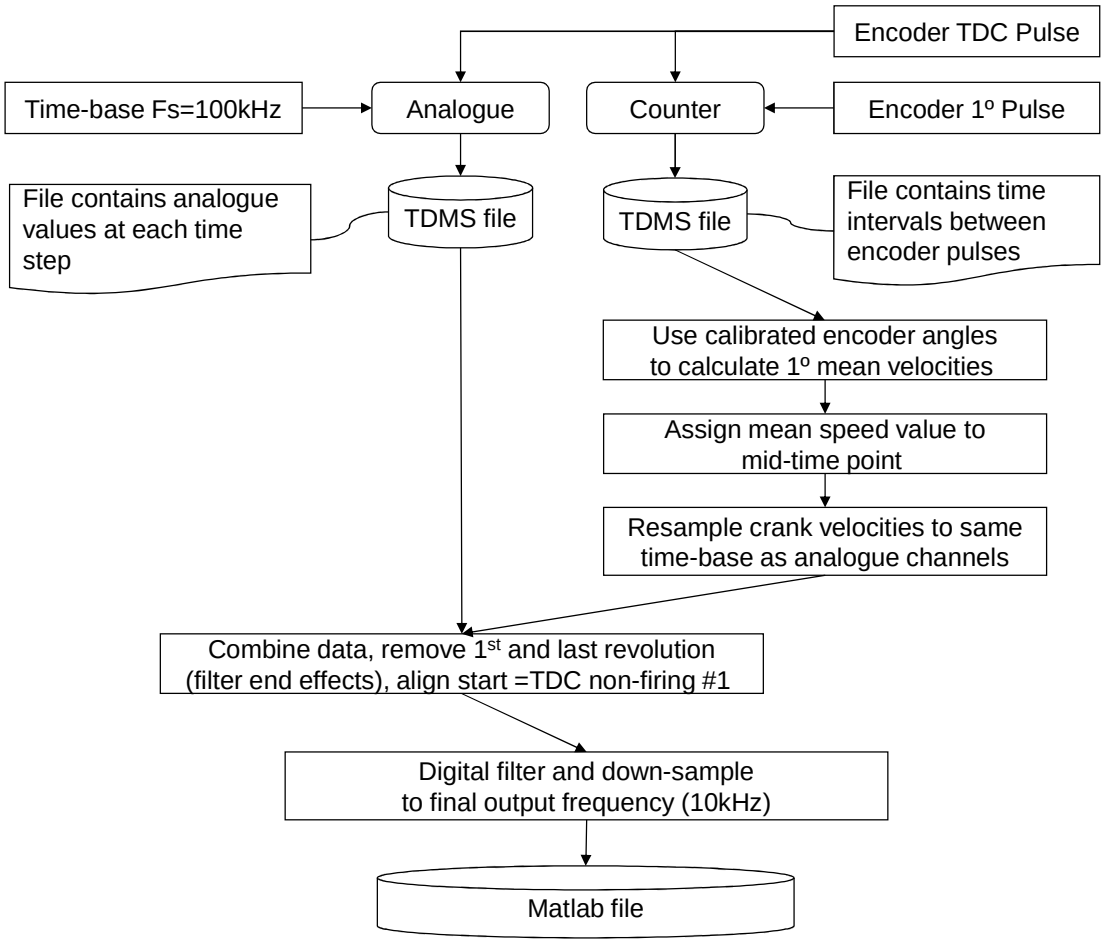


Figure 4.10 – Test data combination schematic

4.6 Test Datasets Acquired

The research programme of which this thesis performs a part includes Jaguar Land Rover Ltd (JLR) as an industrial partner. JLR expressed a particular interest in achieving good cylinder pressure reconstruction performance at low speed, low load operating conditions – these typically being conditions where cylinder pressure is variable and less predictable than high loads where the engine's throttle openings are larger. The small throttle openings necessary at low load result in low volumetric efficiencies and any variations in cylinder filling result in greater percentage changes cycle to cycle and cylinder to cylinder than conditions with larger throttle openings. Partial volumetric efficiencies also result in more variable air motion in the combustion chamber prior to ignition which affects the burn rates and hence the cylinder pressure development.

The less predictable behaviour of cylinder pressure at these low load conditions may offer a greater benefit from advanced combustion control than would result at high loads, if successful reconstruction of cylinder pressure can be achieved. With this consideration, the operating conditions selected to acquire the reference datasets were chosen at 3 speeds – 1000 rev/min, 1500 rev/min and 2000 rev/min and at 3 load conditions – 10 Nm, 20 Nm and 30 Nm. These test conditions give a matrix of data of varying speed and loads summarised in Table 4.4. For each test condition, 2 separate recordings of data were taken, the first to be used for training of ANNs, the second to provide un-seen data to test the trained networks.

In addition to the steady state operating conditions listed above, transient data was also acquired, consisting of slow speed sweeps between 1000 rev/min – 2000 rev/min over a period of 60 seconds. This data has not been employed for the work in this thesis, but is available for future efforts.

Details of the signals acquired from the engine are detailed in Chapter 5, in particular, section 5.3 discusses the variability of the cylinder pressure data across the test matrix.

Engine Speed [rev/min]	Dyno Torque [Nm]		
	10	20	30
1000	✓	✓	✓
1500	✓	✓	✓
2000	✓	✓	✓

Table 4.4 – Matrix of test conditions acquired for engine reference data

Chapter Five

5. PROCESSING OF MEASURED ENGINE SENSOR SIGNALS

After the raw data from the various transducers installed on the engine have been digitised and stored to disk, they then require further processing to make them suitable of use for Artificial Neural Network (ANN) training and testing purposes. This chapter of the thesis describes the signal processing applied to the cylinder pressure signals and the calibration and treatment of crank encoder data to provide acceptable kinematic information. Crank kinematic data is also available from the flywheel inductive probe, and the demodulation of this signal to a useful signal is described, and this data, combined with the crank nose encoder is used to understand the torsional behaviour of the crankshaft. Data from the knock sensor and accelerometer are studied in the time-frequency domain to understand the ranges of importance.

5.1 Cylinder Pressure Signals

Accurate cylinder pressure is an obvious requirement of the test data. If this data is incorrectly scaled or noisy, drifts through the test recording, or is not consistent between cylinders then any attempt to use these pressures as training or validation data for an ANN is flawed from the outset.

This section describes the calibration and ‘pegging’ of the cylinder pressure, the frequency content seen in the signals and the method of combining cylinder pressures from multiple cylinders into a single signal for use in network training.

5.1.1 Pressure Pegging

To achieve the best possible dataset, each transducer / charge amplifier pair was statically calibrated prior to testing using a Druck DPI603. The charge amplifiers were reset immediately prior to each data recording. However, the very nature of the piezo-electric sensors and charge amplifiers means that referencing the output to a known pressure is necessary, known as pegging.

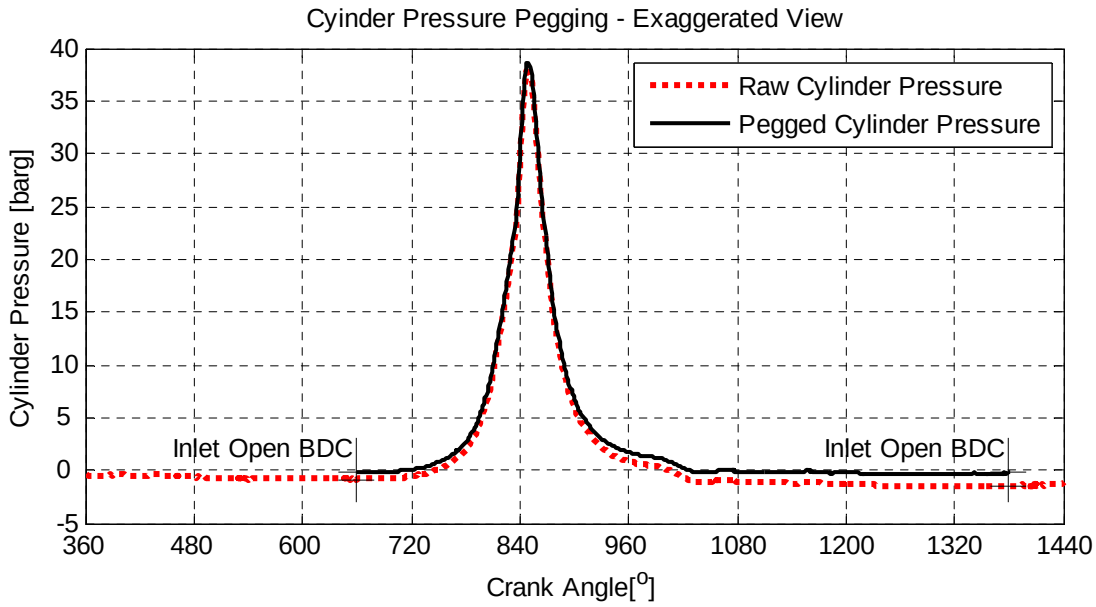


Figure 5.1 – Exaggerated view of cylinder pressure pegging: pressure trace linearly corrected to manifold pressure (-0.25 barg) between consecutive inlet open BDCs (measured data)

Several methods are available, and setting the pressure equal to inlet manifold pressure at Bottom Dead Centre (BDC) when the inlet valves are open is considered acceptable for low speed / load conditions where manifold tuning effects are small [Lee, Yoon, Sunwoo 2007].

As a continuous pressure trace is desired for ANN training, the approach is modified slightly from the per cycle method, and the errors from manifold pressure at two consecutive intake open BDC points are found, and the 720° trace between them is corrected using a linear progression between the errors. This approach avoids the potential step in the corrected pressure curves that would otherwise occur between cycles were a single correction value applied within each cycle. Figure 5.1 shows an exaggerated view of the pegging process.

5.2 Pressure Frequency Content

Understanding the frequency content of the cylinder pressure signal is important. This understanding allows appropriate data acquisition and signal processing bandwidths to be defined, and allows suitable filtering parameters to be applied to noisy signals etc. while preserving important frequency ranges.

Two approaches are used to understand the frequency limits of the cylinder pressure:

i) Fourier analysis is used to study the amplitudes of the time averaged frequency content over 1 pressure cycle.
ii) Low pass filtering to understand the influence of bandwidth on peak pressure magnitude (P_{max}) and position (θ_{max}).

Fourier analysis

A dataset was selected for each test condition, and individual pressure cycles extracted for a crank angle window of 85° before and 140° after TDC firing. For each test point, this yielded 198 cycles of data for each cylinder. The cycles were then analysed via the Fast Fourier Transform (FFT) to provide an estimate of the frequency content averaged across the pressure cycle. Finally the mean of the 198 cycles was found to provide a summary of the information (198 cycles being considered sufficient to provide a confident estimate of the mean).

Figure 5.2 shows the method used to analyse the frequency content of the cylinder pressure. The example shows analysis for cylinder 1 at 1500 rev/min, 10 Nm in the upper plot. The lower plot shows the frequency spectrum of the windowed cylinder pressure for the specific cycle from the upper graph (shown on a semi-log scale – on a linear scale little useful data can be presented), and also for the mean spectrum across all 198 cycles in the dataset.

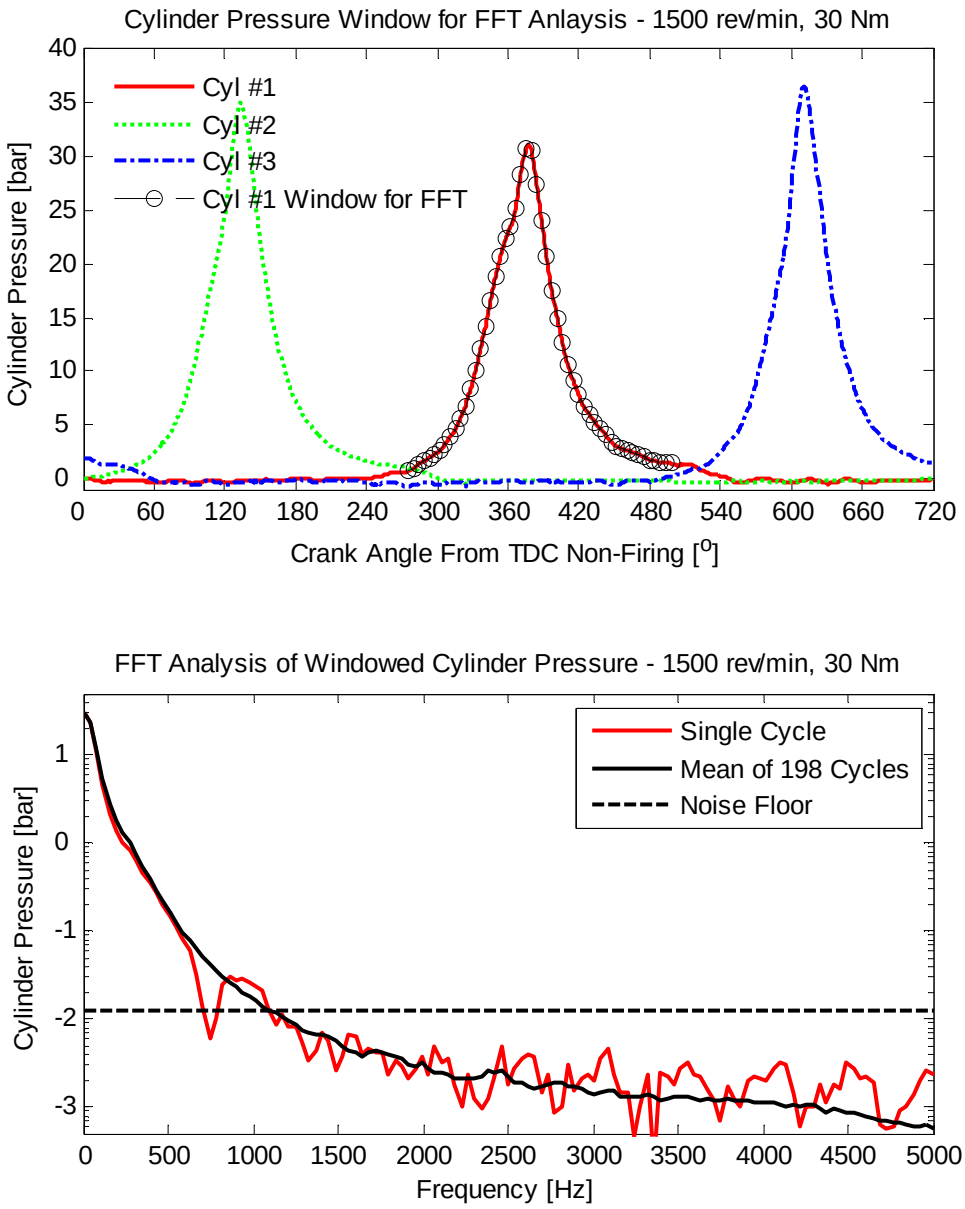


Figure 5.2– Method of FFT analysis for cylinder pressure at 1500 rev/min, 10 Nm Upper: Selection of window around TDC firing for cylinder #1 Lower: Resulting frequency spectrum for individual cycle (red) and mean over 198 cycles (black)

The NI PXI-6133 data acquisition card used to digitise the analogue values from the Ford I3's sensors (see Section 4.4.1) uses a 14-bit analogue to digital convertor (ADC). Such a specification allows for a maximum theoretical dynamic range for the measurements of 84dB. The equivalent noise floor value is shown on the lower graph of Figure 5.5 by the dashed line. This suggests that given the frequency content of the signal, no information above 1000 Hz is valid. In reality, the practical noise floor of the

measurement chain will be significantly above the theoretical values for the ADC itself. Cylinder pressure data at 500Hz is already a factor of approximately 200 less than the very low frequency components, and is likely a realistic limit of the significant frequency content.

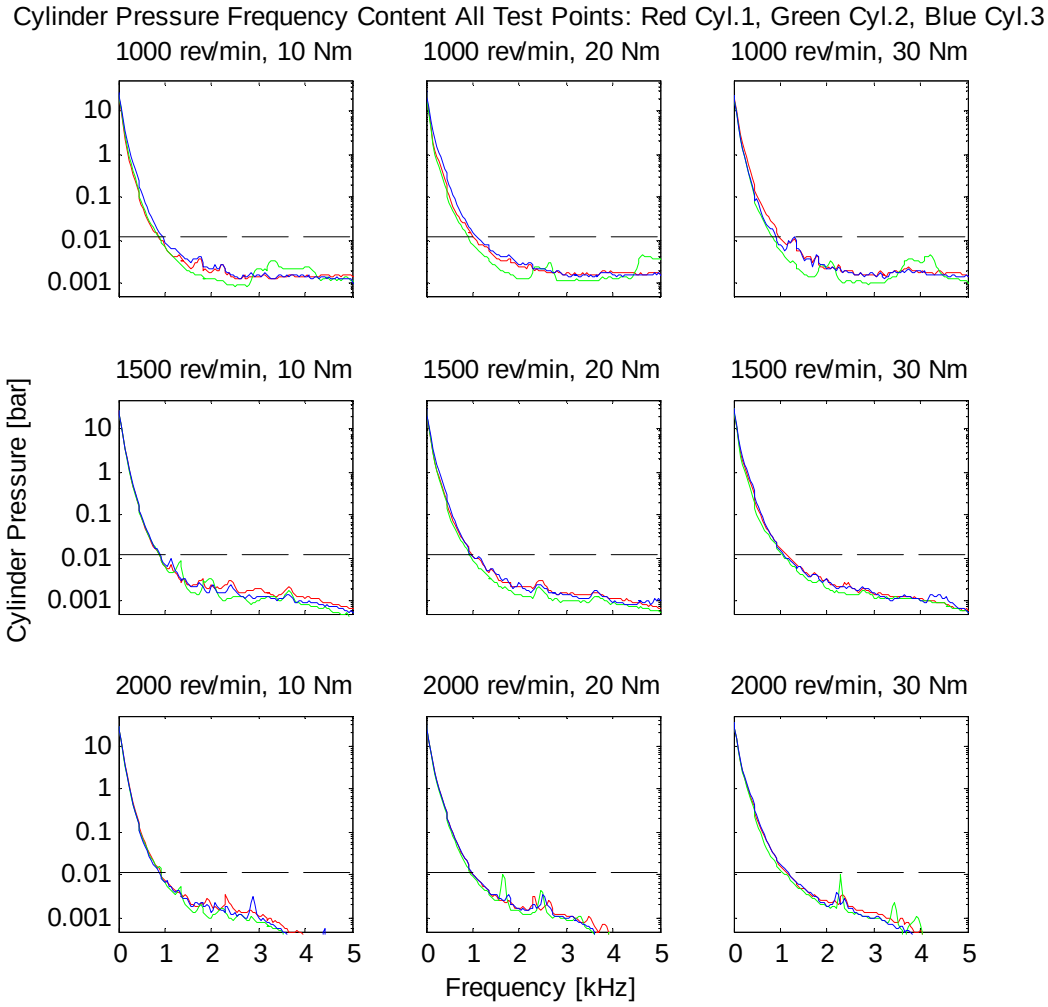


Figure 5.3 – Mean frequency spectra over 198 cycles for each cylinder at each test point, with measurement noise floor shown in dashed black line

Figure 5.3 shows mean spectra over 198 cycles for all 3 cylinders at each of the 9 test points utilised for ANN training. The conclusion drawn above for cylinder 1 at 1500 rev/min, 30 Nm, is also valid for all other test points and cylinders – 1000 Hz represents the highest frequency above the theoretical noise floor of the data acquisition system, and frequencies above 500Hz are a factor of 200 below the low frequency components.

Low pass filtering

Key parameters for the cylinder pressure reconstruction accuracy are the magnitude and position (relative to TDC firing) of peak pressure – P_{max} and θ_{max} respectively. It is important that any reduction of frequency content by low pass filtering does not have a significant effect on these metrics.

A comparison was made by first 'up-sampling' the cylinder pressure data to allow a high resolution of θ_{max}. Up-sampling does not change the frequency content, only the time resolution. The up-sampled data was then progressively low pass filtered, and the effect on P_{max} and θ_{max} evaluated.

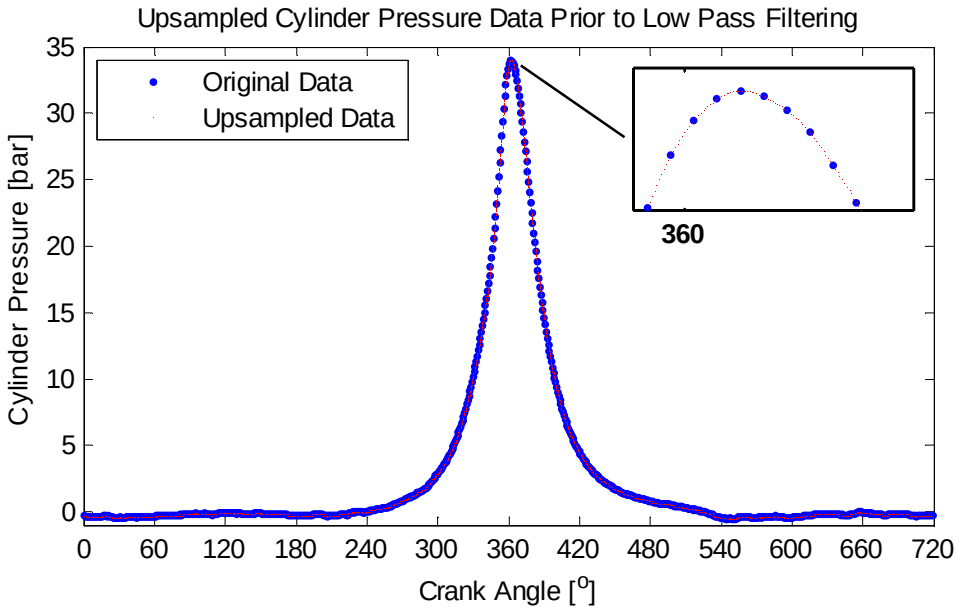


Figure 5.4 – Cylinder pressure data upsampled by a factor of 10 to improve resolution of P_{max} and θ_{max} prior to low pass filtering

Figure 5.4 shows an example of data up-sampled from 10 kHz to 100 kHz allowing a higher resolution of P_{max} and θ_{max}, at this engine speed and sample rate, resolutions of <0.1° crank angle are available.

Figure 5.5 show the effect on the up-sampled pressure trace of low pass filtering to remove high frequency components. All low pass filtering was implemented in a zero phase manner, to ensure filter delay effects were not confused for distortion of θ_{max}. The upper graphs show the low pass filtered pressure trace for 1500 rev/min, 10 Nm. The raw data has the full bandwidth of the standard dataset – sampled at 10 kHz, with a Nyquist frequency of

5 kHz. The lower graph shows the amplitude difference from the unfiltered data.

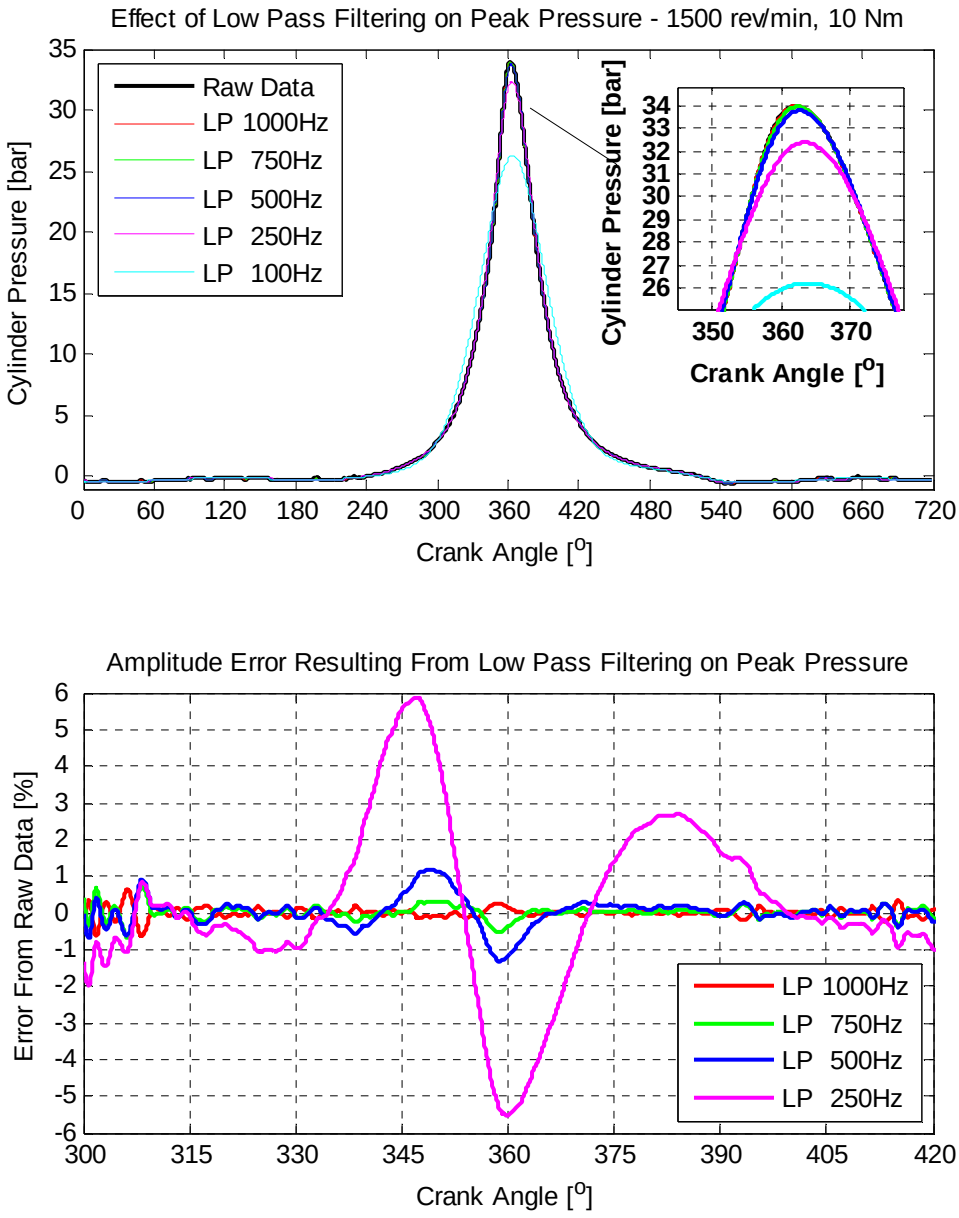


Figure 5.5 – Effect of low pass filtering on cylinder pressure trace at 1500 rev/min, 10 Nm

There is little distinguishable difference between the raw data and the traces low pass filtered at 1000 Hz and 750 Hz. A small difference in P_{max} is seen when the filter cut-off is 500 Hz, and significant reductions of P_{max} result from cut-offs of 250 Hz or below.

Table 5.1 lists the numeric errors resulting from the various low pass filter cut-off frequencies. The conclusion is that low pass filtering at 750 Hz cut-off results in no significant error of P_{max} or θ_{max}

Low Pass Filter Frequency	P_{max} [bar]	P_{max} error [%]	θ_{max} [^{o}]	θ_{max} Error [^{o}]
5000 Hz (raw)	33.98	0	362.1	0
1000 Hz	33.98	0	362.1	0
750 Hz	33.95	- 0.09	362.1	0
500 Hz	33.78	- 0.59	363.0	0.88
250 Hz	32.38	- 4.71	363.9	1.75
100 Hz	26.27	-22.69	363.9	1.75

Table 5.1 – Errors of P_{max} and θ_{max} resulting from low pass filtering of cylinder pressure data at 1500 rev/min, 10 Nm

5.3 Combustion Metrics

If cylinder pressure can be successfully reconstructed, then metrics will be required to compare the predictions against the measured targets, and hence to judge the degree of success of the reconstruction method. Ideally, the full time history of the pressure signal would be re-constructed with minimum error, and if this is achieved, the method would be fully successful. However, there are specific factors within the pressure trace that would be very useful to reconstruct, even if full signal reconstruction is not achieved.

A variety of combustion metrics are employed for gasoline development and calibration, the most basic of these being related to peak cylinder pressure over an engine cycle. The magnitude of peak pressure P_{max} and the crankshaft angle at which it occurs relative to TDC firing θ_{max} are key indicators of basic combustion behaviour. The project industrial partners, Jaguar Land Rover (JLR), would regard consistent reconstruction giving P_{max} amplitude within 4% of measured, and θ_{max} within 1° crank angle of measured to be a useful result. The data would allow adjustment of fuelling volumes and spark timings to balance the work done by each cylinder, and to minimise the variation of individual cylinders cycle to cycle.

The cylinder pressure signals from the datasets described in section 4.6 have been reduced to produce a summary of the variability of performance of peak pressure metrics over 198 engine cycles. Figure 5.6 shows the P_{max} values for each cylinder, within each cycle for each test condition. Figure 5.7 shows similar data for θ_{max}.

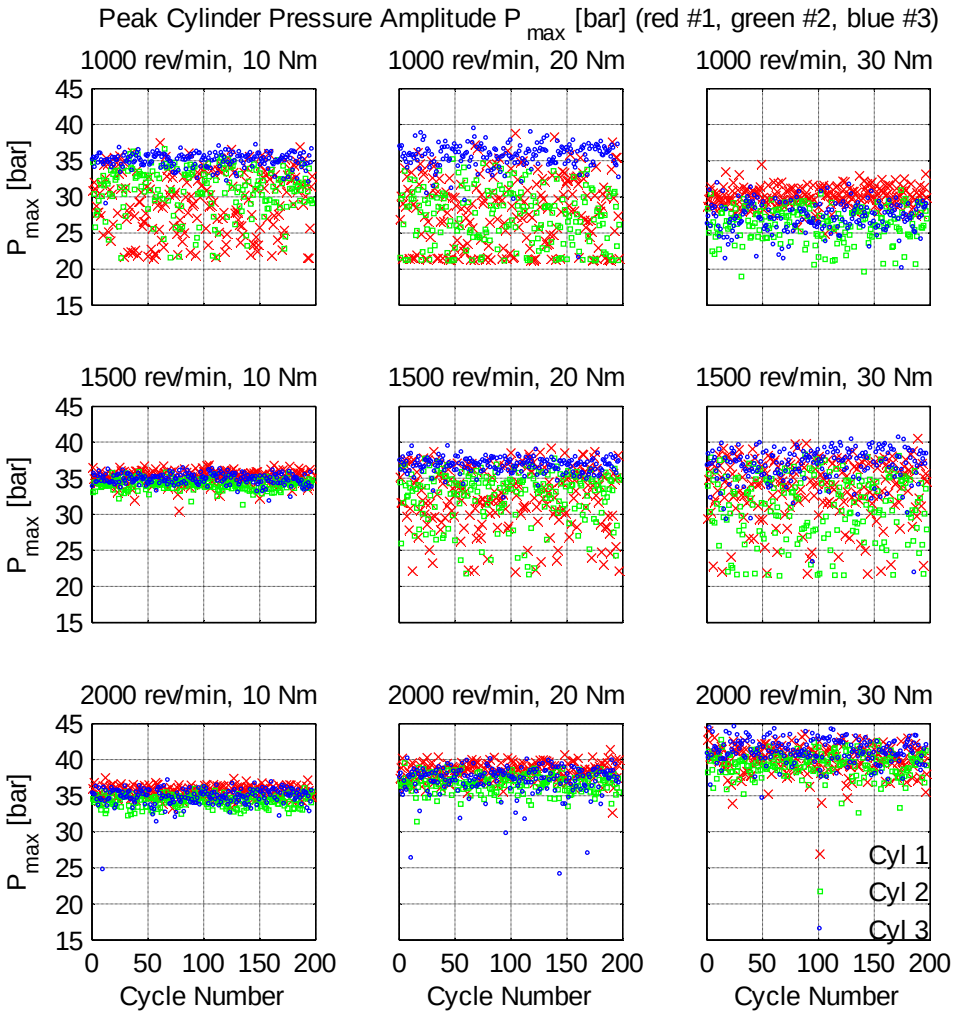


Figure 5.6 – Cylinder to cylinder and cycle to cycle variation of P_{max} for different engine operating conditions

Table 5.2 summaries the statistics of the peak pressure variability, including the measure called coefficient of variation of P_{max}, expressed as a percentage, and defined by equation (5.1) [Stone, 1999 p182]

$$CoV = 100 \bullet \frac{std.deviation\, P_{max}}{mean\, P_{max}} \% \qquad (5.1)$$

These figures and table show the desirable cycle to cycle and cylinder to cylinder variability that will be a strong test for a reconstruction method is present in several of the test conditions.

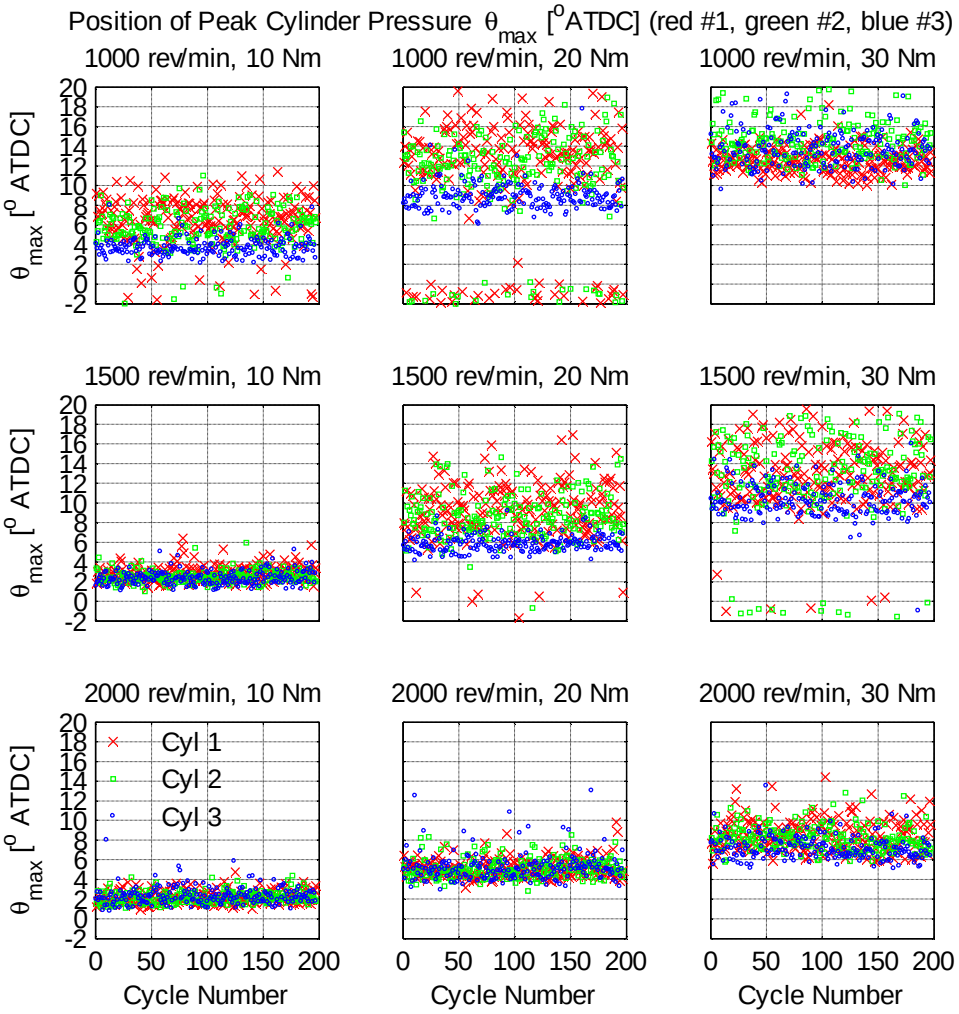


Figure 5.7 – Cylinder to cylinder and cycle to cycle variation of θ_{max} for different engine operating conditions

		P_{max} [bar]									θmax [deg ATDC]					
Speed	Torque	mean			Standard Dev.			CoV [%]			mean			Standard Dev.		
[rev/min]	[Nm]	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1000	10	29.3	30.9	35.0	4.3	3.3	1.0	14.6	10.6	3.0	6.5	5.7	3.7	2.5	1.9	0.9
1000	20	28.0	27.2	35.6	4.8	3.8	1.9	17.1	14.1	5.2	10.8	10.9	8.9	6.0	4.9	1.4
1000	30	30.0	26.2	27.1	1.5	2.3	2.0	4.8	8.7	7.5	12.6	14.6	13.8	1.3	2.2	2.0
1500	10	35.1	34.2	34.8	0.9	0.7	0.7	2.7	2.1	2.1	2.9	2.5	2.3	0.8	0.7	0.7
1500	20	32.4	32.6	36.7	4.2	3.5	1.0	13.0	10.6	2.8	9.2	8.6	5.8	2.8	2.2	0.8
1500	30	33.5	31.0	36.7	4.2	4.4	2.6	12.6	14.2	7.2	13.2	12.8	10.3	3.8	4.5	1.8
2000	10	35.6	34.3	34.8	0.8	0.8	1.2	2.1	2.5	3.5	2.2	2.2	2.3	0.7	0.6	0.9
2000	20	38.4	36.7	37.1	1.2	1.2	2.1	3.1	3.4	5.8	5.2	5.2	5.2	0.9	1.0	1.4
2000	30	40.1	39.3	41.2	1.8	1.6	1.7	4.6	4.2	4.0	8.5	8.2	7.1	1.6	1.3	1.2

Table 5.2 – Summary statistics of peak pressure variations for each cylinder (1,2,3) at each operating condition

5.4 Combination of Pressures Traces from 3 Cylinders to a Single Pressure Signal

When seeking to train an ANN to reconstruct cylinder pressure from crank kinematic data or from cylinder block vibration signals, an issue exists with a multi-cylinder engine that is not present for a single cylinder example.

In the case of crank kinematics, the acceleration of the crankshaft is driven by the resultant torque applied – this torque is generated from 3 cylinder inputs, but the result is a single output of crank acceleration. If cylinder pressure is to be reconstructed for one specific cylinder, then the ANN must create an inverse model which will receive a single input with 3 events per cycle and is required to generate an output with a single event per cycle. If the reconstruction is to be for all cylinders individually, then the network must again be able to separate events within the cycles. This is probably not a great issue for reconstruction methods working in the crank angle domain where individual cylinder events and responses are frequently windowed anyway, but for the time domain approach that is the primary thrust of this work, this is a problem.

A similar problem exists for cylinder block vibration. The vibration signal will have content sourced from inputs from all 3 cylinders, and must separate these into individual cylinder contributions, or learn to ignore 2 of the 3 input events in any cycle if reconstructing for a single cylinder.

This would require the network to develop knowledge of some function of crank position to generate different cylinder pressure outputs through the engine cycle. It seems likely that network training may be simplified if the cylinder pressure could be resolved into a single trace, such that the ANN structure would then have a single output to reconstruct from the homogenous input signals.

The approach used to generate such a single pressure signal is straightforward, and an example is shown in Figure 5.8. The maximum values of the individual overlaid traces for the 3 cylinders for 1 engine cycle (as shown in the upper graph) are assigned to the single combined pressure

trace (as shown in the lower graph). The single combined trace is later used as the single output target for ANN training. The most important features of the pressure trace around TDC firing for each cylinder are not affected by this approach.

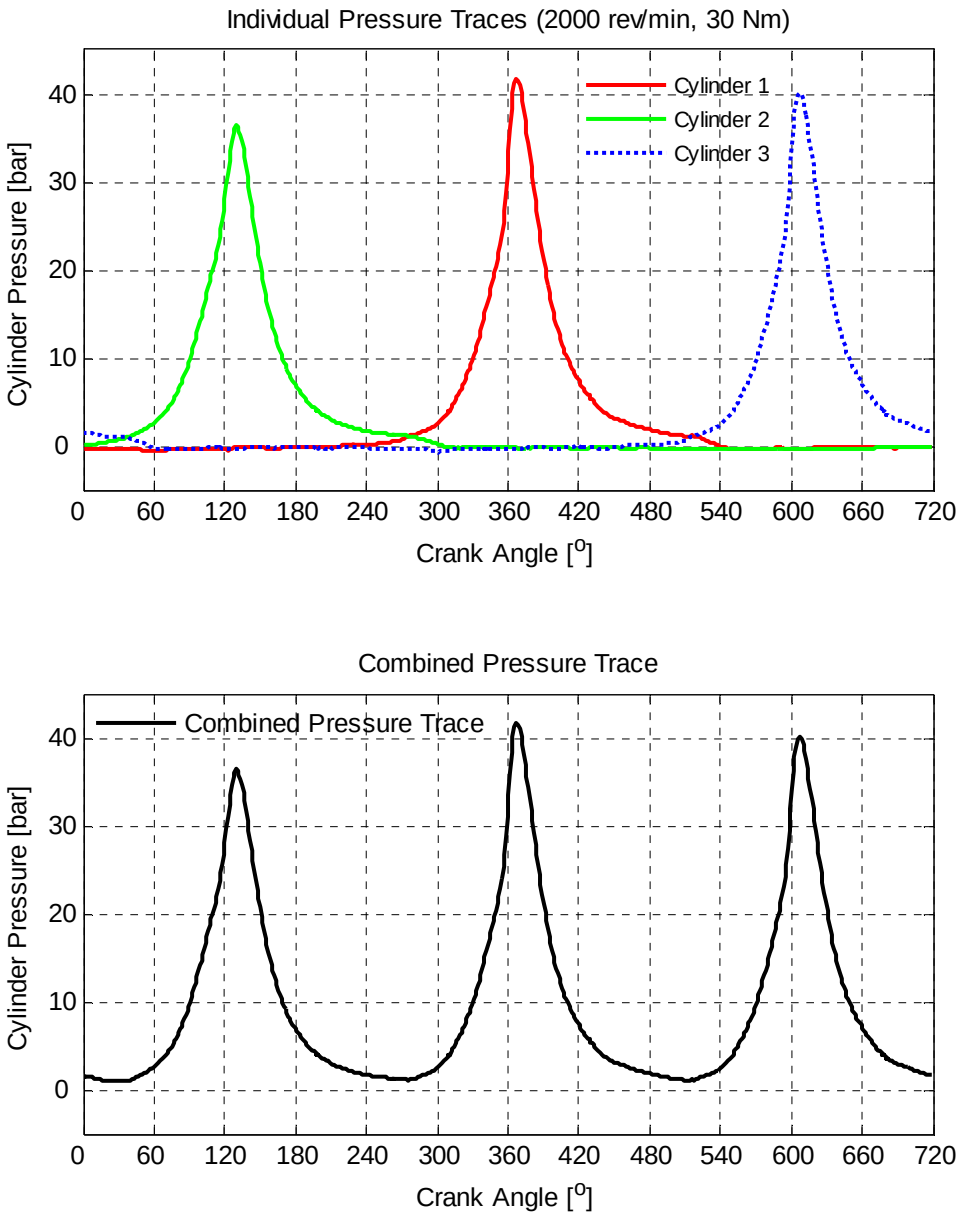


Figure 5.8 – Example of combining 3 cylinder pressure signals to a single trace

5.5 Crankshaft Encoder Data Reduction

5.5.1 Crank encoder calibration and data reduction

Conceptually, the crankshaft encoder is a relatively simple device providing 2 output pulse trains. The first pulse train is a single square TTL pulse of approximately 0.5° duration. The second pulse train is a series of square TTL pulses with rising edges spaced 1° apart, giving 360 pulses per revolution – again the duration of each pulse is approximately 0.5° rotation. Figure 4.5 shows a schematic of the 2 encoder signals.

The rising edge of the first (1ppr) pulse train was carefully aligned at TDC cylinder no. 1. The rising edge of the 'first' pulse of the second (360 ppr) pulse train is coincident with that from the first pulse train. Both pulse trains are connected to 80 MHz counter channels on the PXI-6602. The output from the counter channel provides high time resolution durations between consecutive rising edges for each pulse train. The reciprocal of duration between rising edges gives mean rotation speed – per revolution for the 1ppr signal and per degree for the 360 ppr signal. Use of central finite difference approximation to numerically differentiate the 360 ppr velocity estimate provides crank shaft acceleration.

Initial data acquired from the crankshaft encoder however showed a problem – velocity data showed a high level of noise, ultimately found to be repeatable across every revolution. The amplification of the noisy components resulting from differentiating this velocity signal to acceleration was not acceptable. The upper graph of Figure 5.9 shows an example of crank velocity under engine motored conditions at ≈1000 rev/min. The large low frequency variations (±20 rev/min) seen in the per degree velocity are expected as each piston decelerates towards TDC compression and accelerates after. However the significant high frequency variations are not anticipated, and make the accelerations due to in-cylinder gas pressures indistinguishable among the higher frequency components following differentiation as shown in the lower graph of Figure 5.9. As a primary objective of this work is model gas pressure from crank accelerations, this situation is clearly unacceptable.

Figure 5.9 suggests that the pattern of high frequency noise is repeated over every revolution. Overlaying repeated engine cycles at this motored condition (where higher frequency inputs from combustion, piston slap etc. may be expected to be at a minimum) shows that the degree by degree variations are quite similar.

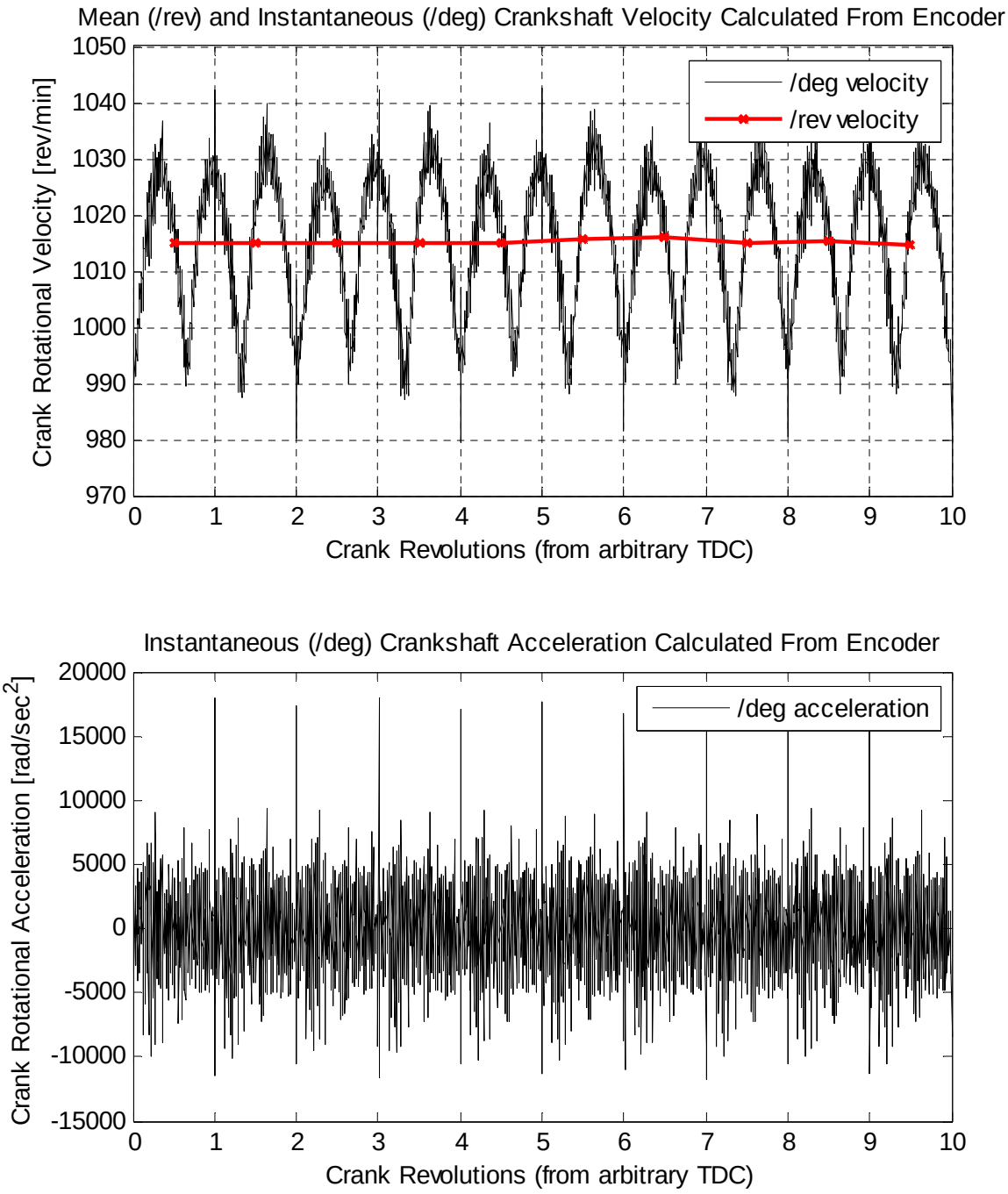


Figure 5.9 – Raw encoder data – noisy crank kinematic data from motored engine recording ≈1000rev/min
Upper: Raw crank rotational velocity
Lower: Acceleration from encoder signal

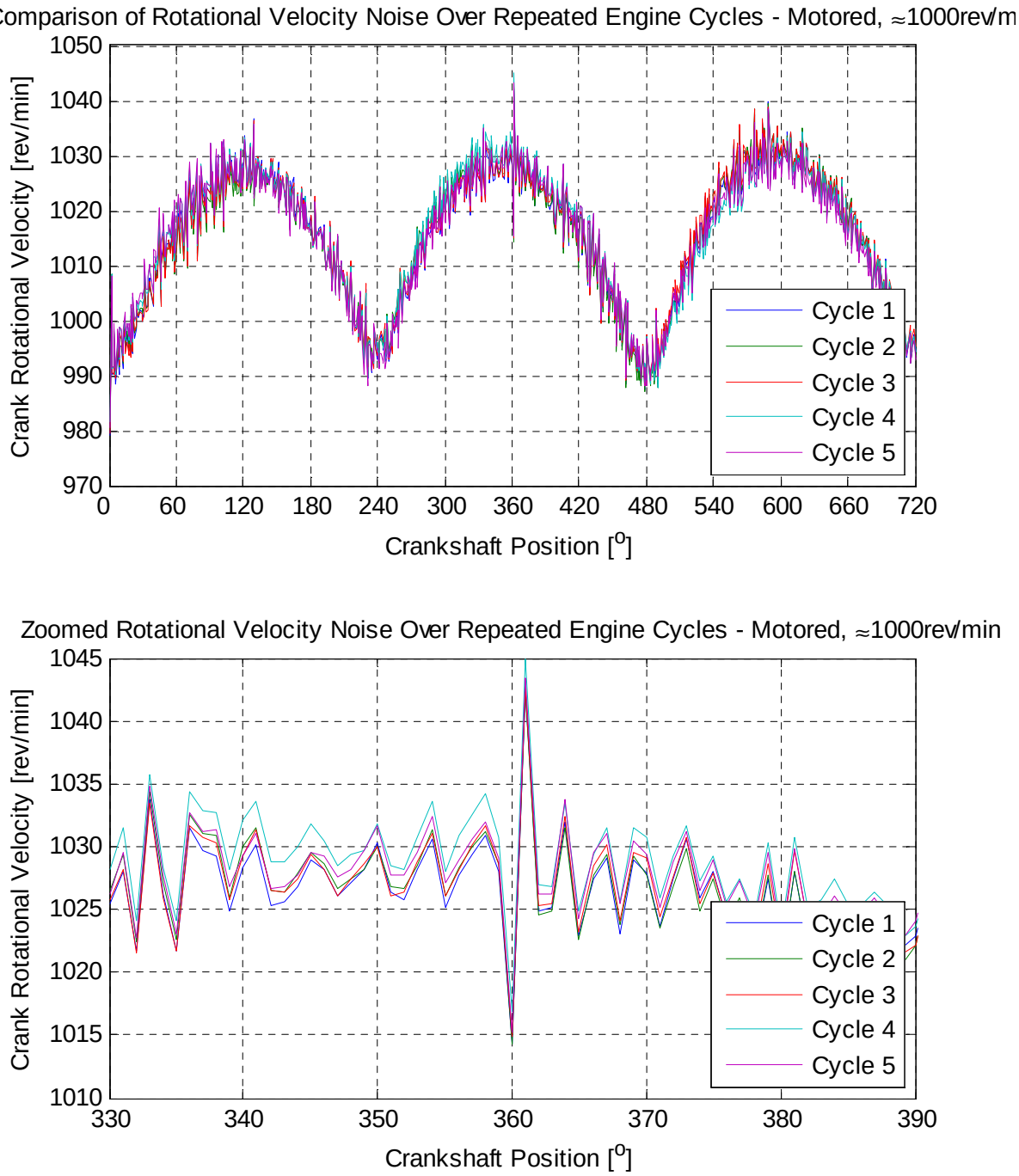


Figure 5.10 – Overlaid cycles of crankshaft velocity data showing repeatable degree to degree error
Upper: Data over 720°
Lower: Zoomed detail over 60° centred on TDC

Figure 5.10 show the 5 engine cycles (10 revolutions) previously discussed overlaid. The upper graph shows the full cycles, the lower is zoomed to a 60° window around TDC exhaust stroke on cylinder 1. This figure does clearly suggest that the degree to degree variations are repeated on every

engine rotation. There also appears to be a consistently larger correction at the TDC location (360°).

The hypothesis to explain this data is that the degree markers in the encoder are not sufficiently accurately spaced to allow degree by degree crank acceleration to be extracted. Every degree pulse appears to have a repeatable error, and there is a larger correction as the encoder completes 360° of rotation and the cumulative angular error is effectively reset.

The encoder in question is an industry standard item, used quite universally for crank angle input to combustion analysis systems. In that case however, it is being used as a position indicator rather than for velocity or acceleration, and the small errors in the angles are not deemed problematic. When differentiated, their relative magnitude increases dramatically.

To resolve this issue, an approach to calibrate the encoder was developed. If the encoder could be rotated at a constant speed, then the degree pulse rising edges should be equally spaced in time. Any deviation from the expected equal time spacing could be measured, and a correction calculated – effectively identifying to a high resolution exactly how many degrees each pulse represented rather than assuming 1.000° for each.

By accurately measuring the time interval between TDC positions (i.e. 360° CA) on a disc with either constant or slowly-varying speed ω, this data can be used to fit a simple polynomial (e.g. a cubic-spline) for interpolation purposes, which allows the time intervals t_{int} between ideal 1° crank angles to be interpolated. Designating the measured time interval between the actual TTL pulses as t_{TTL} then the error ε_i can be estimated for i=1°,…,360° as described by equation (5.2):

$$\varepsilon_i = \omega_i (t_{\text{int}} - t_{TTL}) \tag{5.2}$$

However, achieving a constant rotation speed is not straightforward, all readily available motors proved to have some degree of torque variation and hence speed ripple within each revolution, and attempts to drive the encoder at a constant speed were not successful. Instead, an inertia disc was

machined and installed to the encoder drive face. The steel disk had an inertia value of approximately 475×10^{-6} kgm^2, and allowed sufficient energy to be stored to enable 10 to 15 revolutions of the encoder as it coasted down to stationary under internal friction only.

An assumption was made was that internal friction varied as a smooth function of rotational speed and was not significantly variable within each revolution. Both 1 ppr (TDC) and 360 ppr (degree) pulse trains were acquired to the 80MHz counters of the NI acquisition system.

The 1 ppr is assumed to have no revolution to revolution error – the pulses generated must be physically 360° apart as they are generated by the same optical slot in the encoder disc – any timing inconsistencies delivered by the electronics in the encoder cannot be accounted for by this calibration method. The cumulative times for rotations (for the 1 ppr signal) are shown in Figure 5.11 with a cubic spline curve fitted through these times.

This cubic polynomial is then used to model the time at which 1° spaced pulses should appear. The differences between these modelled times and the counter measured times of the 360 ppr signal gives the encoder error that must be re-calibrated. Figure 5.12 shows the calibrated angles in degrees for each pulse from the 360 ppr pulse train.

Repeated revolutions give very similar results, and a cyclic mean over 10 revolutions is used as the final calibration values. Table 5.3 lists the calibrated angles per pulse for the 360 ppr signal as used for further encoder signal processing throughout this thesis. When the calibrated angles for each of the 360ppr pulses have been derived, then those angular values can be used to calculate the crank rotational velocity instead of the nominal 1° values.

Figure 5.13 shows a comparison of crankshaft rotational velocity calculated with the original assumption that each pulse occurs at 1° spacing (black line), with that calculated from the calibrated encoder angles (red line) as detailed in Table 5.3. Clearly, the calibrated values result in a significantly smoother rotational velocity trace, with much reduced sample to sample variation, and

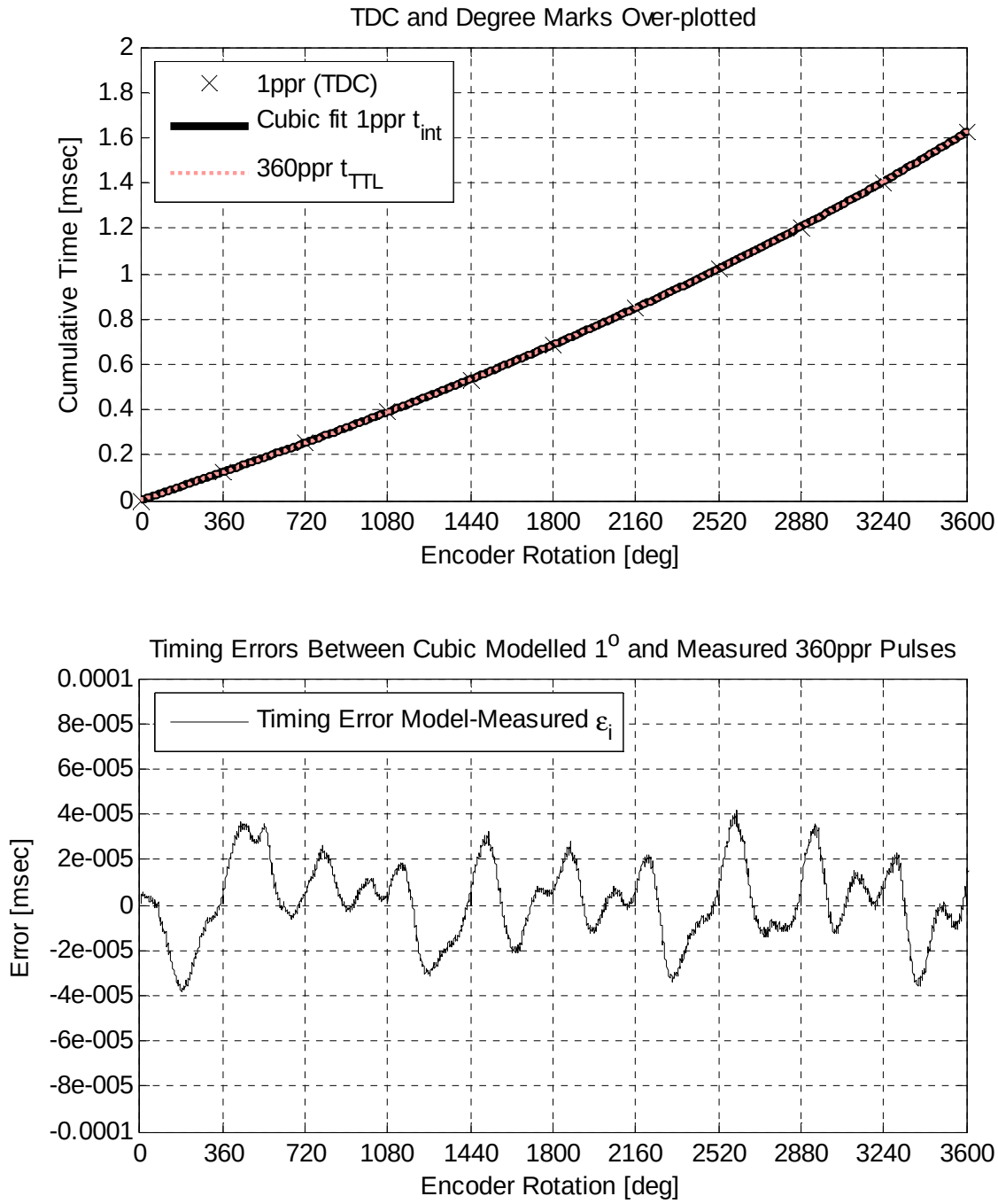


Figure 5.11 – Encoder calibration data, 10 revolutions -
Upper: Overlaid modelled and measured 1° intervals
Lower: Comparative timing errors

the elimination of the large correction at 0° encoder angle (TDC cylinder no.1 for the crankshaft).

Numerically differentiating the calibrated rotational velocity signal to give rotational acceleration gives a very significant improvement in the signal to noise ratio of the resulting acceleration data as shown in Figure 5.14.

However, the acceleration signal remains somewhat noisy, a perennial problem of numerically differentiating any measured time dependent signal.

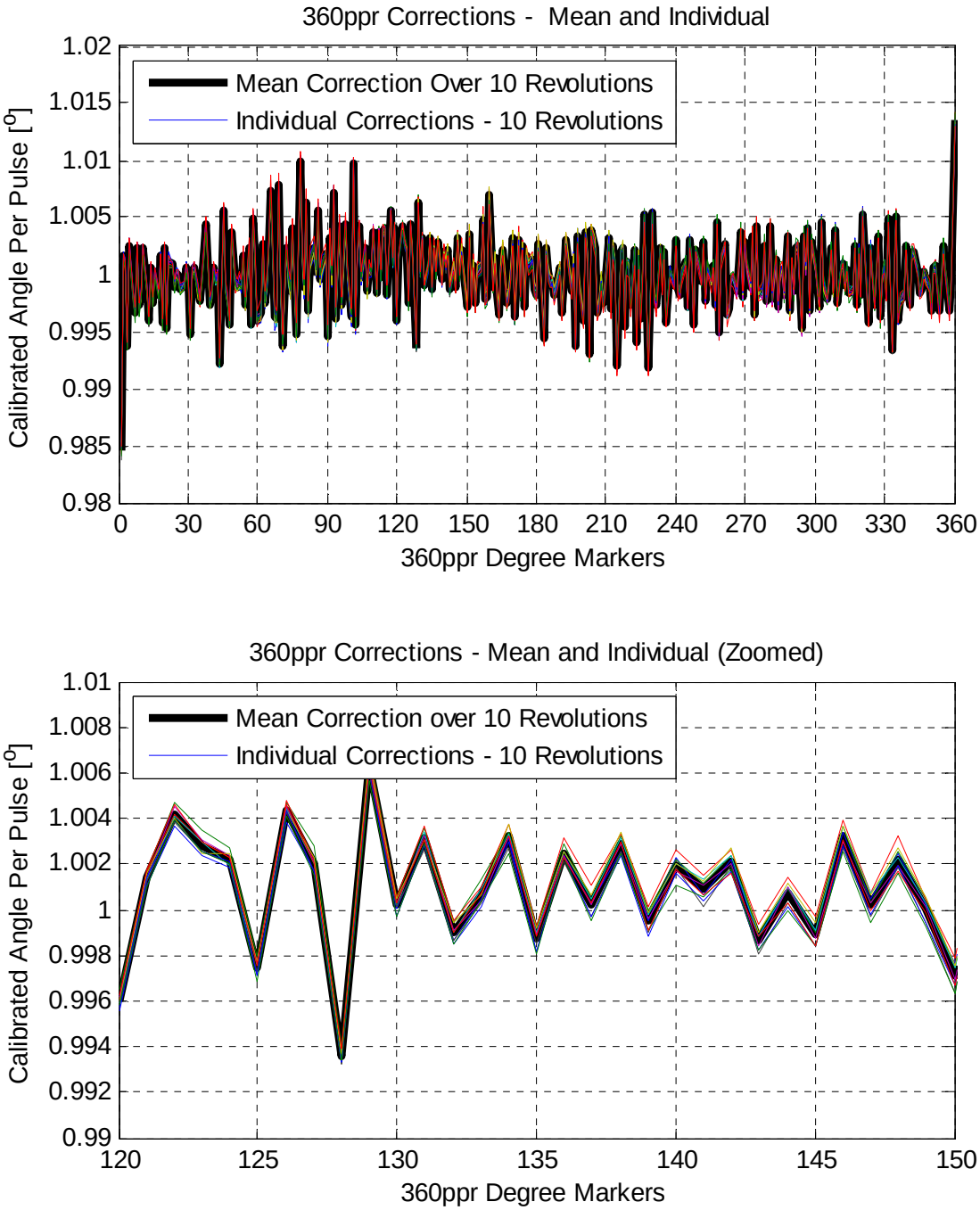


Figure 5.12 – Calibrated encoder degree per pulse (360ppr) – Upper: Mean compared to 10 individual revolutions, Lower: Zoomed detail comparing mean to 10 individual revolutions over 30° rotation

Crank Encoder Calibration Values: Degree Values Per Pulse From 360ppr Signal															
Pul. No.	Cal Degs	Pul. No.	Cal Degs	Pul. No.	Cal Degs	Pul. No.	Cal Degs	Pul. No.	Cal Degs	Pul. No.	Cal Degs	Pul. No.	Cal Degs	Pul. No.	Cal Degs
1	0.9846	46	0.9988	91	1.0032	136	1.0025	181	1.0027	226	1.0003	271	1.0008	316	0.9986
2	1.0016	47	0.9977	92	0.9998	137	1.0002	182	1.0008	227	1.0053	272	0.9989	317	1.0025
3	1.0009	48	0.9956	93	1.0072	138	1.0029	183	0.9945	228	0.9918	273	1.0033	318	0.9973
4	0.9937	49	1.0037	94	0.9962	139	0.9995	184	1.0023	229	0.9980	274	0.9965	319	1.0020
5	1.0025	50	0.9991	95	1.0043	140	1.0019	185	1.0003	230	1.0054	275	1.0042	320	0.9982
6	0.9981	51	0.9989	96	0.9974	141	1.0009	186	0.9993	231	0.9981	276	0.9984	321	1.0053
7	0.9966	52	0.9993	97	1.0008	142	1.0021	187	0.9994	232	0.9972	277	0.9985	322	0.9989
8	1.0023	53	0.9990	98	1.0045	143	0.9987	188	0.9981	233	1.0023	278	1.0024	323	0.9958
9	0.9973	54	1.0016	99	1.0001	144	1.0007	189	0.9995	234	0.9991	279	0.9977	324	1.0023
10	0.9981	55	0.9973	100	0.9962	145	0.9989	190	1.0006	235	1.0020	280	0.9994	325	0.9997
11	1.0022	56	1.0024	101	1.0097	146	1.0032	191	0.9966	236	0.9959	281	1.0043	326	0.9973
12	0.9988	57	0.9957	102	0.9955	147	1.0002	192	0.9993	237	0.9982	282	0.9979	327	1.0034
13	0.9960	58	1.0049	103	1.0012	148	1.0022	193	1.0031	238	1.0004	283	1.0013	328	0.9962
14	1.0006	59	1.0017	104	1.0042	149	1.0000	194	0.9998	239	0.9990	284	0.9972	329	0.9997
15	0.9975	60	0.9959	105	0.9997	150	0.9971	195	0.9986	240	1.0030	285	1.0001	330	1.0008
16	1.0002	61	0.9966	106	1.0023	151	1.0035	196	1.0012	241	0.9987	286	1.0005	331	0.9981
17	0.9982	62	1.0027	107	0.9985	152	0.9978	197	0.9937	242	0.9982	287	0.9983	332	1.0049
18	1.0016	63	0.9975	108	1.0007	153	1.0008	198	1.0034	243	0.9999	288	1.0034	333	0.9934
19	0.9957	64	0.9993	109	1.0002	154	0.9975	199	0.9988	244	1.0011	289	0.9967	334	1.0006
20	1.0024	65	1.0023	110	1.0038	155	1.0002	200	0.9960	245	0.9971	290	1.0010	335	1.0051
21	0.9953	66	1.0073	111	0.9984	156	1.0007	201	1.0038	246	1.0031	291	0.9986	336	0.9960
22	1.0012	67	0.9962	112	1.0036	157	1.0048	202	1.0014	247	1.0011	292	1.0024	337	0.9977
23	1.0009	68	0.9961	113	0.9998	158	0.9978	203	0.9930	248	0.9955	293	1.0025	338	0.9983
24	0.9990	69	1.0078	114	1.0038	159	0.9981	204	1.0039	249	1.0021	294	0.9952	339	1.0024
25	1.0001	70	0.9987	115	1.0040	160	1.0069	205	1.0023	250	1.0006	295	1.0034	340	0.9987
26	0.9999	71	0.9938	116	0.9983	161	0.9989	206	0.9967	251	0.9993	296	1.0007	341	0.9973
27	0.9987	72	1.0024	117	1.0055	162	1.0014	207	0.9976	252	1.0029	297	1.0041	342	1.0019
28	0.9990	73	0.9998	118	1.0011	163	0.9982	208	0.9995	253	0.9977	298	0.9969	343	1.0019
29	1.0006	74	0.9986	119	1.0041	164	0.9965	209	0.9975	254	1.0006	299	1.0028	344	1.0003
30	0.9997	75	1.0040	120	0.9960	165	1.0007	210	0.9991	255	0.9992	300	1.0006	345	0.9983
31	0.9947	76	0.9992	121	1.0014	166	1.0022	211	1.0033	256	1.0007	301	0.9987	346	0.9987
32	1.0003	77	0.9947	122	1.0041	167	0.9976	212	0.9971	257	0.9973	302	0.9971	347	1.0002
33	1.0005	78	1.0099	123	1.0028	168	0.9983	213	1.0025	258	1.0045	303	1.0046	348	0.9996
34	0.9987	79	1.0015	124	1.0022	169	1.0000	214	1.0010	259	0.9949	304	0.9991	349	0.9989
35	0.9976	80	0.9988	125	0.9975	170	1.0032	215	0.9920	260	1.0013	305	0.9978	350	0.9968
36	0.9996	81	1.0062	126	1.0044	171	0.9962	216	1.0020	261	1.0027	306	0.9998	351	1.0006
37	1.0019	82	0.9968	127	1.0020	172	1.0031	217	1.0023	262	0.9964	307	1.0023	352	1.0007
38	1.0044	83	1.0013	128	0.9936	173	0.9985	218	0.9955	263	0.9970	308	0.9985	353	0.9988
39	0.9974	84	1.0024	129	1.0063	174	1.0024	219	1.0011	264	0.9996	309	1.0038	354	0.9968
40	0.9983	85	1.0003	130	1.0002	175	0.9972	220	1.0005	265	0.9998	310	0.9975	355	1.0025
41	1.0006	86	1.0055	131	1.0031	176	1.0016	221	0.9973	266	0.9985	311	1.0007	356	1.0002
42	0.9995	87	0.9996	132	0.9990	177	1.0008	222	1.0022	267	1.0007	312	1.0009	357	0.9986
43	0.9999	88	1.0007	133	1.0006	178	1.0012	223	0.9940	268	1.0037	313	0.9990	358	0.9968
44	0.9921	89	1.0018	134	1.0032	179	0.9987	224	1.0015	269	0.9981	314	1.0004	359	0.9993
45	1.0056	90	0.9945	135	0.9987	180	0.9982	225	0.9961	270	1.0029	315	0.9984	360	1.0136

Table 5.3 – Individual calibrated degree values for encoder 360 ppr pulse train, mean values from 10 calibration revolutions

Figure 5.15 shows the effect (over one crank revolution) of filtering the crankshaft acceleration signal derived from the calibrated encoder data to reduce the noise using filters with cut off frequency at 90th, 45th, 27th and 18th engine orders. A clear reduction in signal noise is seen with the 18th order cut-off filter. Filter cut-off frequency may be an important factor in preparing data for input to an ANN.

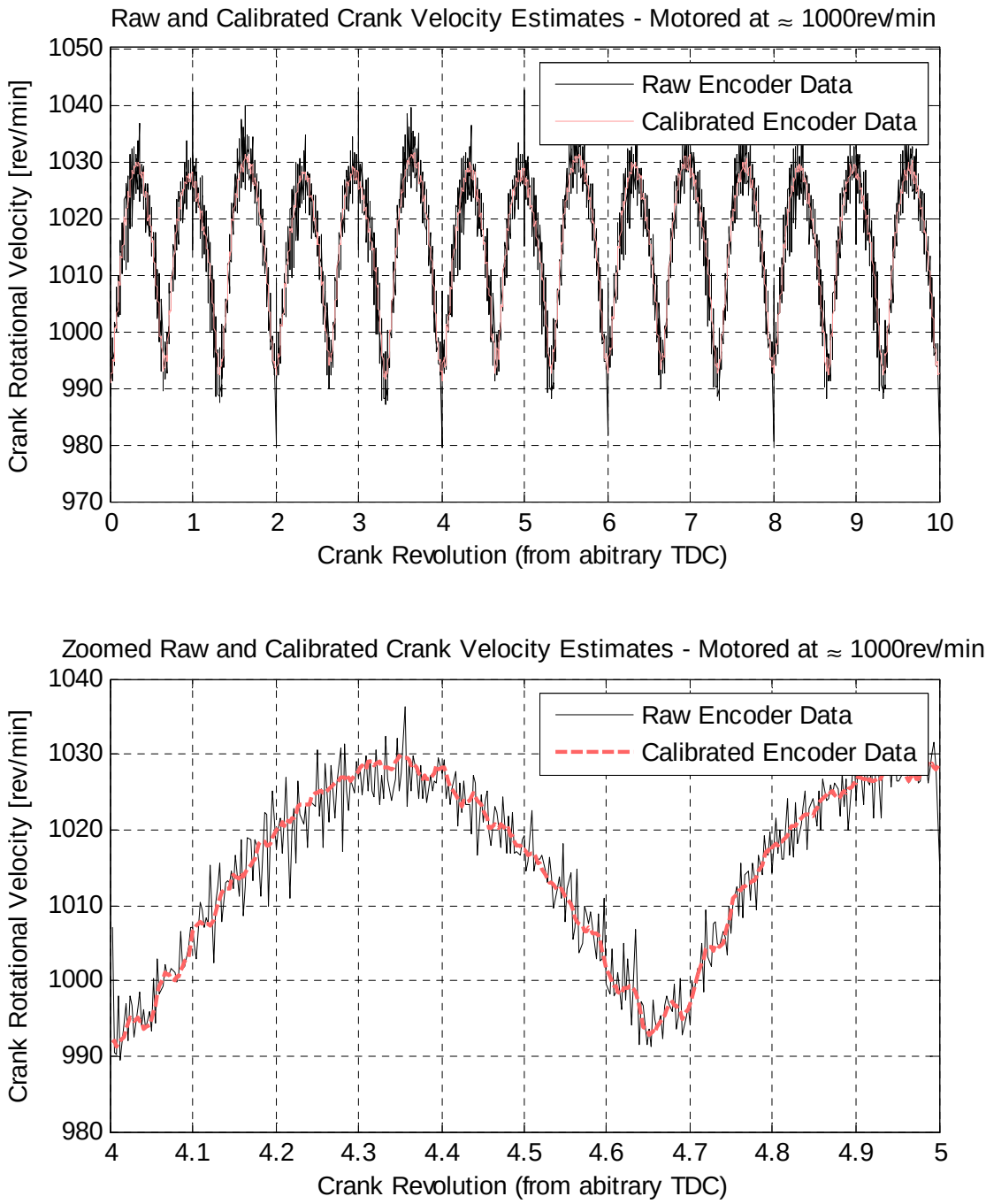


Figure 5.13 – Crankshaft rotational velocity using raw and calibrated encoder signals, engine motored at ≈1000 rev/min –
Upper: Comparison over 10 encoder revolutions
Lower: Zoomed detail over one revolution

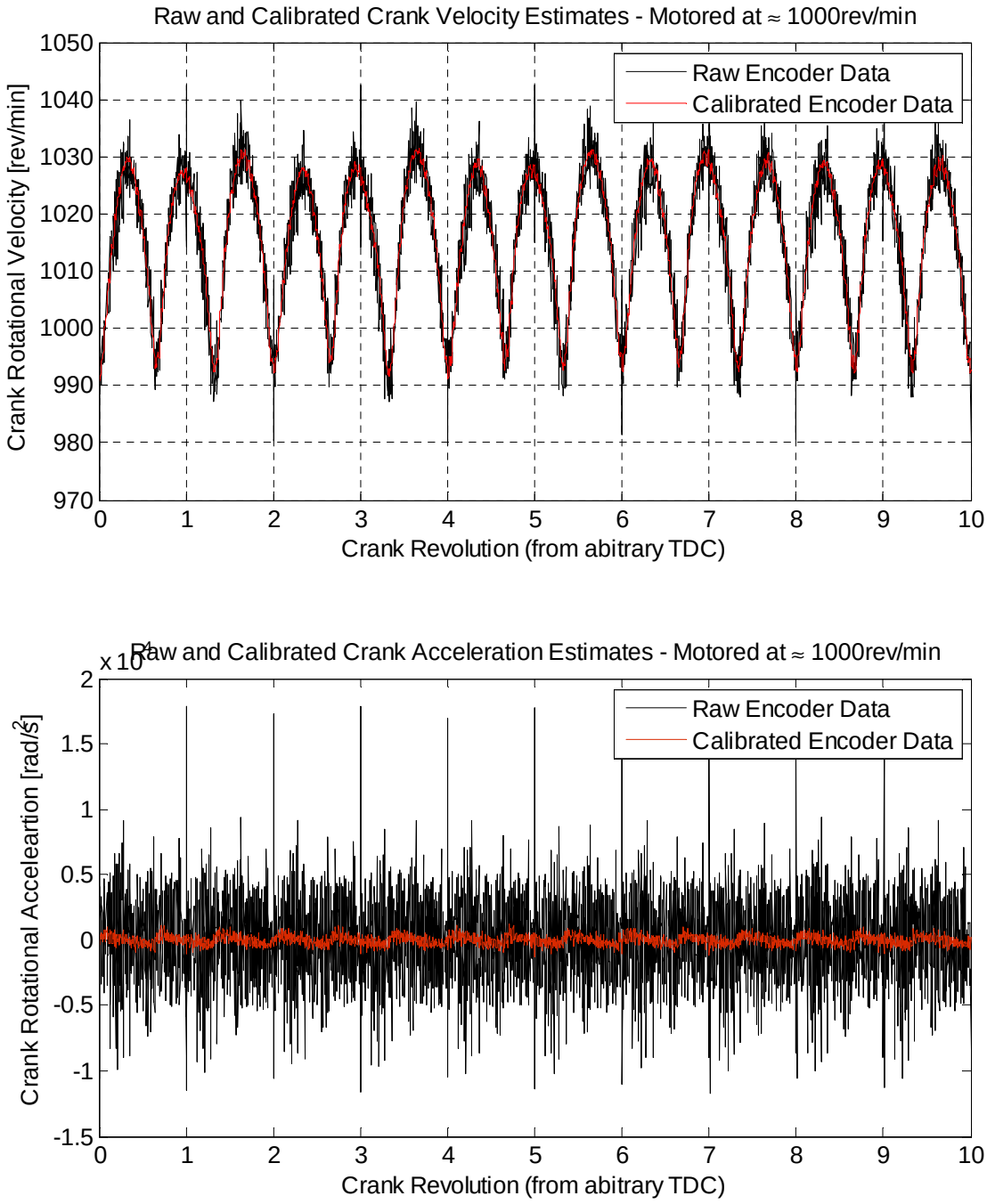


Figure 5.14 – Comparison of numerical differentiation of un-calibrated and calibrated crank velocities to rotational acceleration
Upper: Un-calibrated versus calibrated crankshaft velocities
Lower: Acceleration signals derived from un-cal'd and cal'd signals

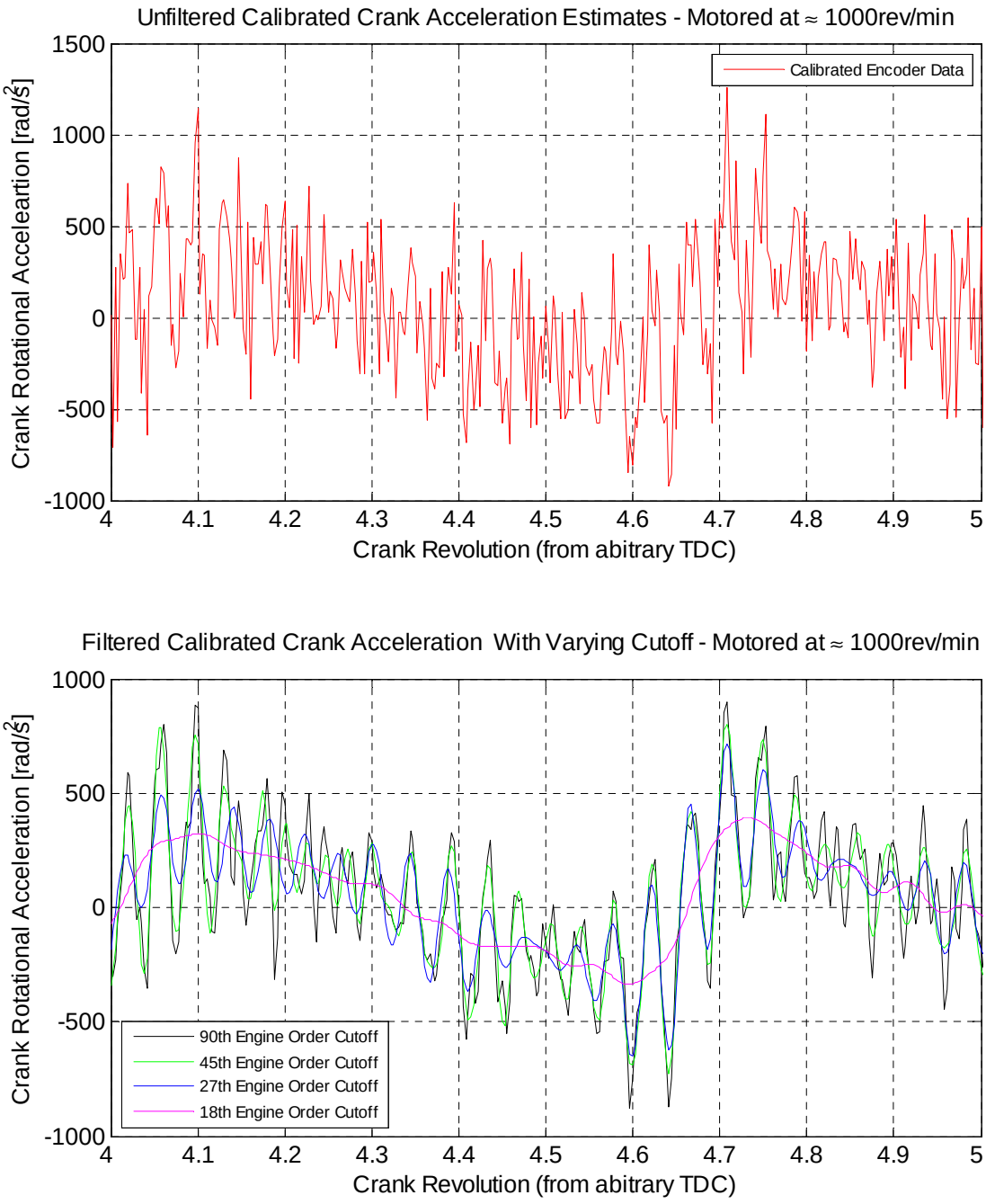


Figure 5.15 – Comparison of filter cut-offs to reduce noise in crankshaft acceleration derived from calibrated encoder data -
Upper: Crankshaft acceleration derived from calibrated encoder data
Lower: Crankshaft acceleration data filtered at decreasing engine orders

5.5.2 Higher Encoder Resolutions

The Kistler optical encoder specification includes the capability of higher than 1° (360 ppr) angular resolution:

0.5° Resolution

0.5° resolution should be feasible by using the falling edge of the TTL pulse train as well as the rising edge. The pulses are nominally 0.5° wide, and hence configuring the counter to measure time elapsed between each rising /falling and falling/rising pair should provide crank kinematic data resolved to 0.5°. A set of motored data was acquired in this manner, but was found to be noisy, and the calibration method described in section 5.5.1 failed to produce an acceptable result. The reason for this could not be clearly established, as each pair of 0.5° pulses summed to match data triggered on rising edges only (i.e. 1° spacing) to within 1 count of the 80 MHz counter – equivalent to 1.25 nanoseconds, or $75°x10^{-6}$ at 1000 rev/min. It is suspected that the signal conditioning electronics are not as accurate with respect the falling edge of the pulse as they are for the rising edge – any inaccuracy at this edge would result in both 0.5°intervals being corrupted.

0.1° Resolution

0.1° resolution should be feasible with the use of the Kistler Type 2614A4 pulse multiplier, which provides an output of 3600 ppr. The method of increasing the pulse output by a factor of ten over the optical slits cut in the encoder disk was not clear from the Kistler literature. A set of test data was acquired with the 1° pulse train connected to one NI counter, and the 0.1° pulse train connected to a second counter. The result of this test was the discovery that the 9 pulses available between the 1° pulses appeared to be at equal time intervals. This suggests that the pulse multiplier may work by reading the time between 2 consecutive 1° marks, then forward synthesising 0.1° pulses spaced at 0.1 of that time. This effectively assumes that the velocity for the next 1° is unchanged from the preceding, measured degree and consequently contains no useful additional data regarding crankshaft acceleration.

As a result of the above investigations, all crank encoder data within this thesis use values taken at 1° intervals (360 ppr), calibrated according to the method described in section 5.5.1.

5.6 Extraction of crank velocity data from inductive probe targeted on flywheel teeth

Inductive probes are commonly installed in production engines to provide crankshaft position and speed information for the Engine Control Unit (ECU) to drive various control functions such as ignition and injection timing etc. Access to the signal from the Ford I3's built-in sensor was not available, so an additional sensor was installed targeted radially on the flywheel teeth. Figure 5.16 show the installation of the probe on the engine.

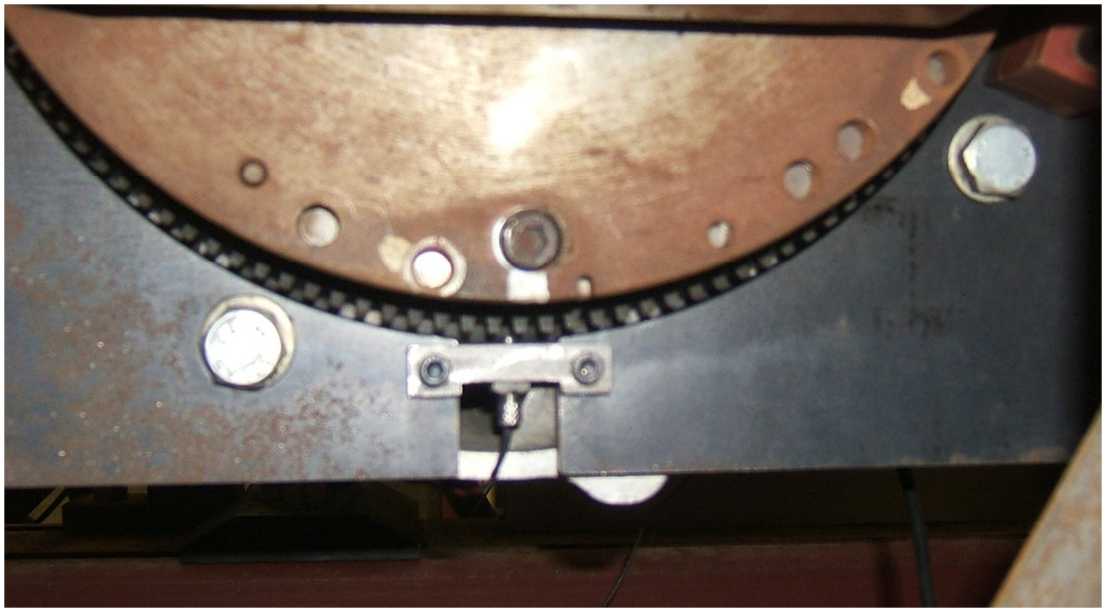


Figure 5.16 – Low cost inductive probe targeted on flywheel teeth

Data from the probe (along with other engine sensors) was recorded during speed sweeps across the engine's operating range from 1000-6000 rev/min. Two separate load conditions were tested:

Minimum load condition where the engine was stabilised at 1000 rev/min with no dynamometer load applied, and the engine speed increased by opening the throttle steadily to increase engine speed linearly to

6000 rev/min over a period of approximately 60 seconds – effectively the load on the engine is internal friction only.

Maximum load condition where the engine was stabilised at 1000 rev/min with the throttle fully open, and the speed controlled by the dynamometer in speed set-point mode, the speed set-point was then steadily increased to 6000 rev/min over a period of approximately 60 seconds – the engine load hence being maximum at all speed points.

The purpose of including this sensor measurement from the engine was 2 fold:

i) To understand if data from such a low cost sensor could be used to provide crank kinematic data to drive the cylinder pressure reconstruction process – the one off sensor cost at 2012 prices was below £10, and hence in high volumes could be considered affordable for production engine installation whereas the cost of the Kistler optical encoder would not be appropriate for production.

ii) To provide crank kinematic data from the opposite end of the crankshaft to Kistler optical encoder – potentially allowing flexible crank dynamics to be studied and their influence on cylinder pressure reconstruction understood. To satisfy this case, the data used in this section is from a sweep across the full speed range 1000-6000 rev/min as any crank flexible modes are likely only to be excited at higher engine speeds (and hence higher cylinder firing frequencies).

The signal generated from the sensor when the engine is running is a distorted sine wave (the exact signal shape is dependent on the gear tooth form, proximity of the inductive probe tip to the teeth and flywheel eccentricity) - with frequency equal to the flywheel teeth passing frequency. The signal amplitude varies with the proximity of the probe tip to the teeth – a closer mounting result in higher output, and on engine speed – higher engine speed results in higher output.

These latter points require some care in preparation to ensure the probe is installed at a distance from the crown of the gear teeth such that at maximum engine speed the signal amplitude does not exceed the maximum input level for the data acquisition system otherwise clipping of the signal will result with consequent distortion and loss of information. Alternatively an analogue integrating circuit can be included in the sensor which results in relatively constant amplitude as the integration reduces amplitude as frequency increases, however such a circuit would add some cost to the system.

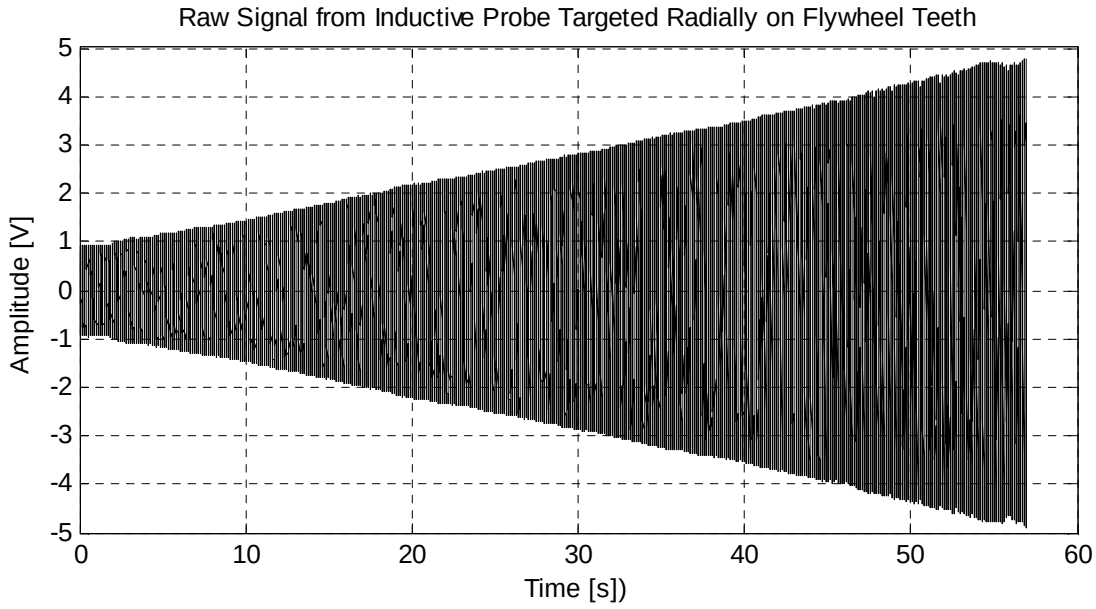


Figure 5.17 – Raw signal from inductive probe targeted on flywheel teeth over the engine speed range 1000 rev/min – 6000 rev/min, maximum load

There is frequently some degree of amplitude modulation of the signal over each engine rotation due to non-concentricity of the gear teeth crowns. Figure 5.17 shows the raw signal from the sensor over the full engine speed range from 1000 rev/min (time: 0 sec) – 6000rev/min (time: 55 sec), the amplitude increase with speed is clearly seen. Figure 5.18 shows a small zoomed segment of the data (approx. 20 teeth passing) at 1000 rev/min, the distorted sinusoidal nature of the signal is clear.

The purpose of the signal from the inductive probe is to provide a second source of crank kinematics resolved within each engine cycle. There are two

methods available to extract the crank kinematic data – zero crossing and frequency demodulation.

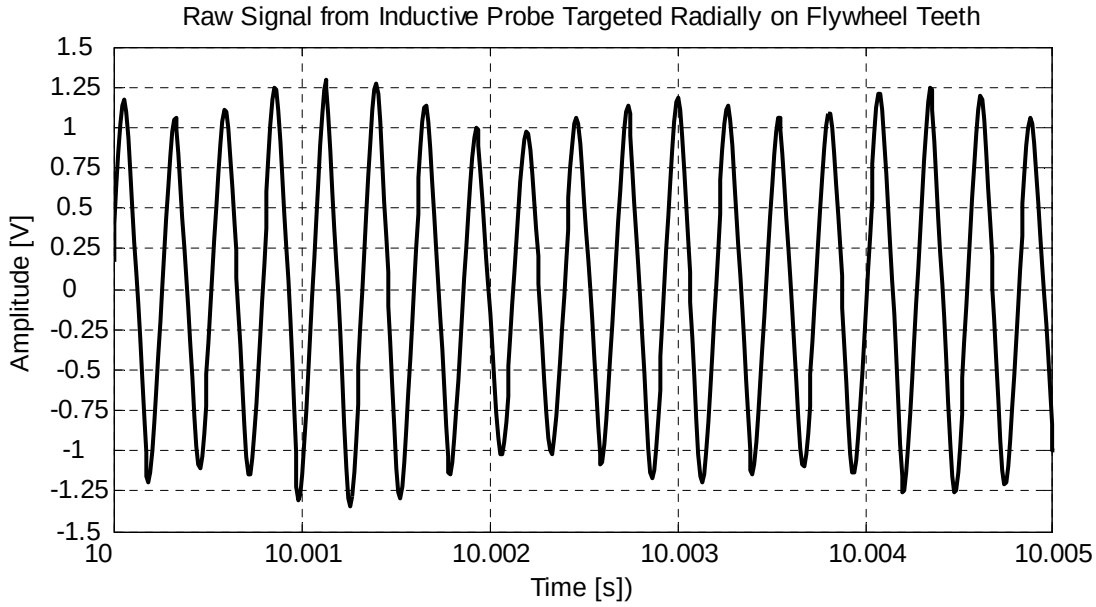


Figure 5.18 – Inductive probe signal zoomed at 1000 rev/min, full load

Zero crossing

The zero crossing approach treats the signal from the flywheel in a similar manner to the data taken from the encoder, finding the time taken to complete one tooth passing cycle, dividing into the angular displacement associated with one tooth to give a rotational velocity estimate for that angle. To avoid the effects of any non-symmetry in the positive and negative slope of the probe signal, only zero crossings from the positive slopes are considered.

In the case of the inductive probe however, the signal is acquired to an analogue input of the data acquisition system rather than a counter, and hence the time resolution available is limited to that used as an analogue sample frequency. This resolution limit gives rise to a velocity estimate that increases in uncertainty with mean speed and with the number of flywheel teeth.

The uncertainty also increases with reducing sample rate:

$$\omega_t = \frac{1}{N} \bullet \frac{60}{t_t}$$

Where: ω_t is the mean speed over 1 tooth passing event

N is the number of teeth on the flywheel

t_t is the measured time for one tooth passing event

With 135 teeth on the Ford I3 flywheel, and an analogue sampling rate of 10 kHz, the potential speed measurement error due to mis-timing the zero crossing point by one analogue sample is shown in Table 5.4

Mean Crank Speed [rev/min]	±Error Due to 1 Sample [rev/min]
1000	22
2000	86
3000	190
4000	330
5000	505
6000	714

Table 5.4 – Error in per flywheel tooth speed estimate due to mis-timing of zero crossing by 1 analogue sample

Figure 5.19 shows the differences in various methods of estimating crank shaft rotational velocity. It is clear that resolution of the estimates from flywheel teeth zero crossing at 10 kHz sample rate are poor at high speeds.

Figure 5.20 shows the same data as Figure 5.19, but with detail at low engine speed (1000 rev/min). Here the data shows that with 10x up-sampling, the speed estimates are comparable to those from the encoder degree marks. Figure 5.21 shows similar detail, but now at 5600 rev/min where the degree by degree speed estimate from the flywheel teeth signal using zero crossing processing are unacceptably noisy for either the raw data or the 10x up-sampled data. It is concluded that estimating in-cycle crank speed from the flywheel teeth signal using zero crossing calculation must be

used with great caution and consideration of the speed under test and the analogue sample rate is critical.

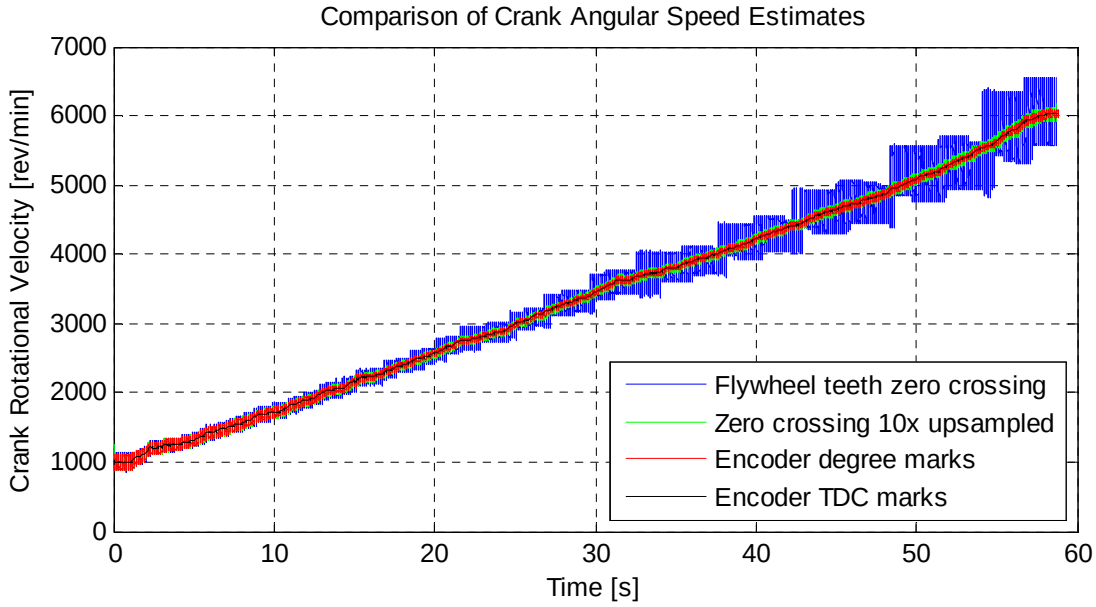


Figure 5.19 – Comparison of crank angular speed estimates across the engine speed range at full load from:
Encoder TDC marks (black), Encoder degree marks (red),
Flywheel teeth zero crossings (blue) and
Flywheel teeth zero crossings after 10x upsampling (green)

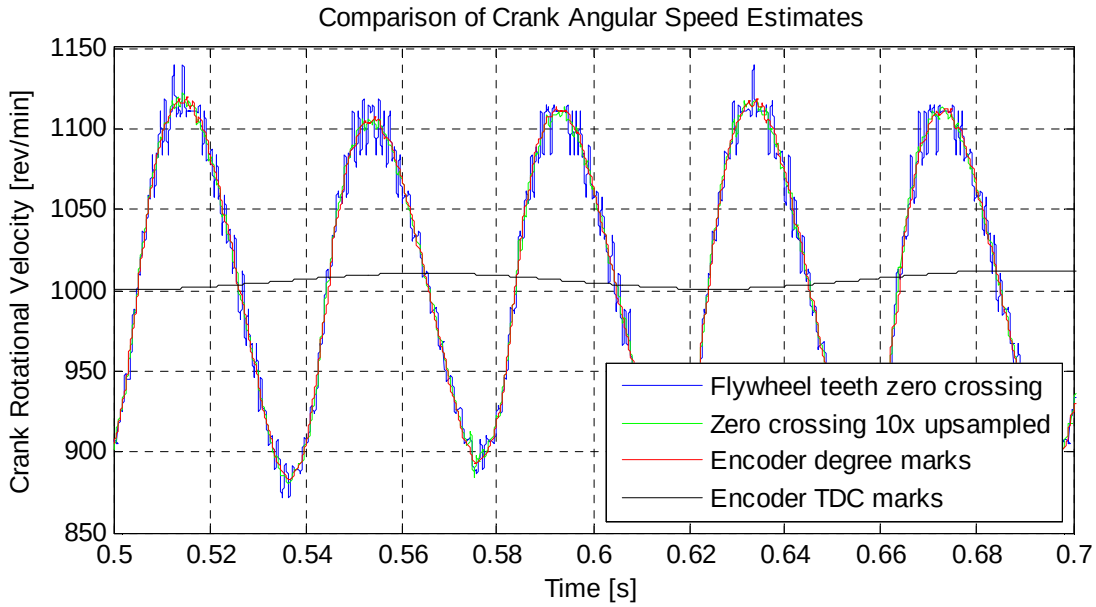


Figure 5.20 – Comparison of crank angular speed estimates at ≈1000 rev/min, full load:
Encoder TDC marks (black), Encoder degree marks (red),
Flywheel teeth zero crossings (blue) and
Flywheel teeth zero crossings after 10x upsampling (green)

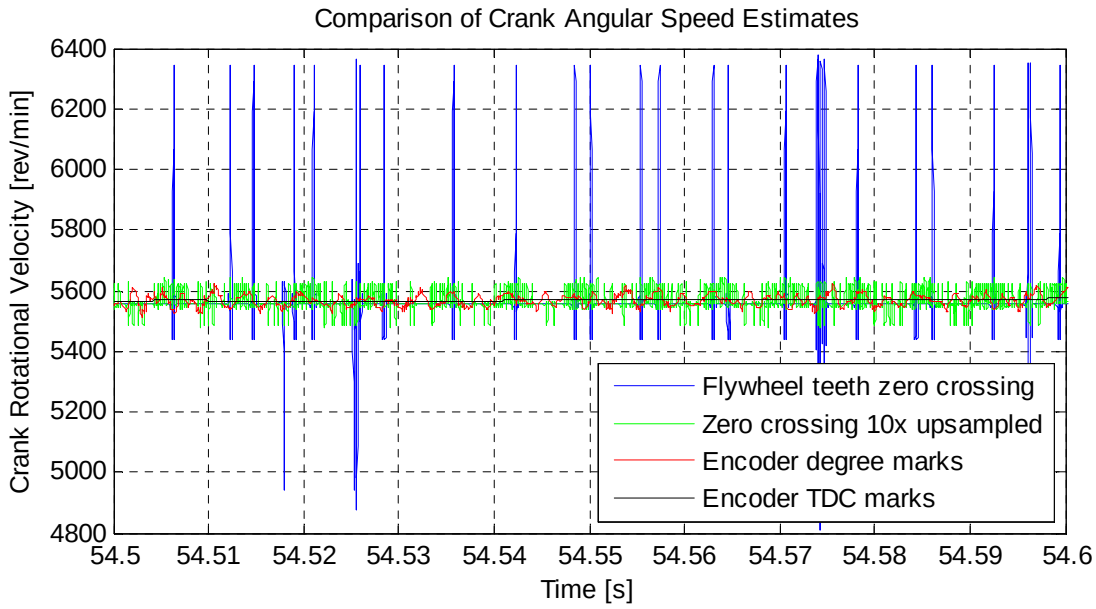


Figure 5.21 – Comparison of crank angular speed estimates at ≈5600 rev/min, full load: Encoder TDC marks (black), Encoder degree marks (red), Flywheel teeth zero crossings (blue) and Flywheel teeth zero crossings after 10x upsampling (green)

Frequency Demodulation

If the flywheel were rotating at a constant angular velocity, then the signal produced by the teeth passing the inductive probe would be a distorted sine wave. The degree of distortion is partly due to inconsistencies in the tooth form, to flywheel eccentricity and to differences in the probe's output at maximum (crown) and minimum (root) tooth heights. However the sine wave is also modified by changes in the angular velocity though the engine cycle – the tooth passing sine wave becomes a carrier wave and is frequency modulated by the velocity changes, hence frequency demodulation processing may be applied to the signal to extract the velocity information.

The method chosen to recover the instantaneous velocity information from the flywheel teeth signal is phase demodulation using the Hilbert transform, as described in [Tuma 2002], [Zho and Ouyang 2009].

The Hilbert transform *H(t)* changes *cosines* to *sines* and vice versa:

$$H\left[sin(t)\right] = -cos(t)$$
$$H\left[cos(t)\right] = sin(t)$$

hence effecting a 90° phase shift of the signal. An analytic version $x_a(t)$ of the original real signal $x(t)$ is generated:

$$x_a(t) = x(t) + iH\left[x(t)\right]$$

The instantaneous frequency is the rate of change of phase of the analytic signal, computed by numerical differentiation of the unwrapped phase angle of the analytic signal. This can be constructed quite simply in MATLAB (the Hilbert transform is a built in function), but the operation depends on the FFT in the step to generate to 90° shifted signal. The raw signal, scaled to rev/min is shown in Figure 5.22, compared to the encoder degree pulse generated speed trace. A high degree of noise can be seen in the signal, occurring at tooth passing frequency. It is necessary to filter out this contamination; the green curve in Figure 5.22 shows the instantaneous speed after filtering at 50% of tooth passing frequency.

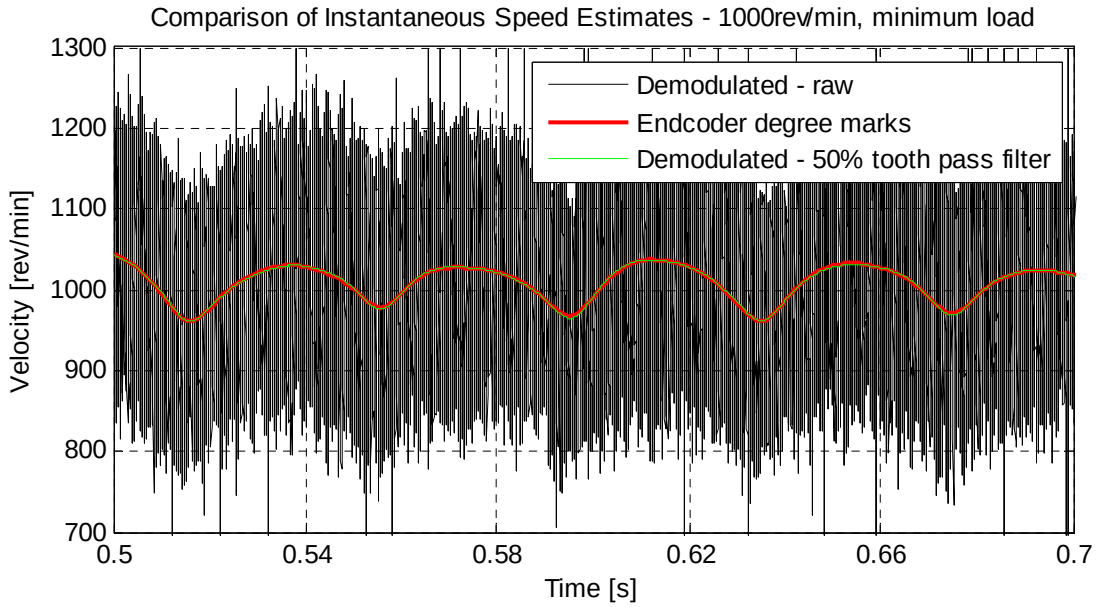


Figure 5.22 – Instantaneous crankshaft velocity estimate calculated from Hilbert demodulation of flywheel teeth signal with and without tooth passing frequency filter, compared to encoder degree pulse estimated velocity

The use of the FFT causes a problem for non-stationary data such as a transient speed sweep from an operating engine. To resolve this, the flywheel tooth signal is processed in crank revolution length pieces. The engine speed range of 1000-6000 rev/min is traversed in approximately 55

seconds, and hence the mean speed change over one revolution is approximately 5 rev/min at 1000 rev/min (less at higher mean speeds). This is small relative to the intra-revolution speed change and hence the signal is a reasonable approximation to stationary. The piecewise calculation also allows tooth passing frequencies to be filtered from the resulting signal with a fixed frequency cut-off filter.

An alternative to this piecewise processing is resampling of the signal to the angle domain prior to computing the Hilbert transform – the velocity fluctuations then calculated are in the engine order rather than the time domain, and the tooth passing filter can be fixed at a constant engine order. The data can be transformed back to the time domain after the demodulation processing is complete if required.

It is concluded that demodulation of the signal from the inductive probe targeted on the flywheel teeth using the derivative of phase from the signal's Hilbert transform followed by suitable low pass filtering to eliminate tooth passing frequencies results in an acceptable measure of instantaneous crank velocity. The data extracted from the flywheel probe compares closely to the data from the encoder at low engine speeds, but shows some differences at high speeds. Figure 5.23 compares the 2 estimates at minimum and maximum crank speeds.

At 6000 rev/min, the mean estimates are close, but the encoder shows larger amplitude and higher frequency fluctuations than the flywheel probe data. This is expected – at some point through the engines speed range, the firing frequency will exceed the crankshaft first torsional mode, and above that point the shaft can no longer be considered rigid – hence the two ends of the crank can be expected to show some instantaneous difference of kinematics. The direction of the difference shown in Figure 5.23 is also expected – the flywheel has significantly higher inertia than the crank nose and could be expected to have the lower peak amplitude of velocity fluctuation. Section 5.7 looks at the torsional flexibility of the crankshaft as identified by the kinematic measurements at the crank nose and flywheel in more detail.

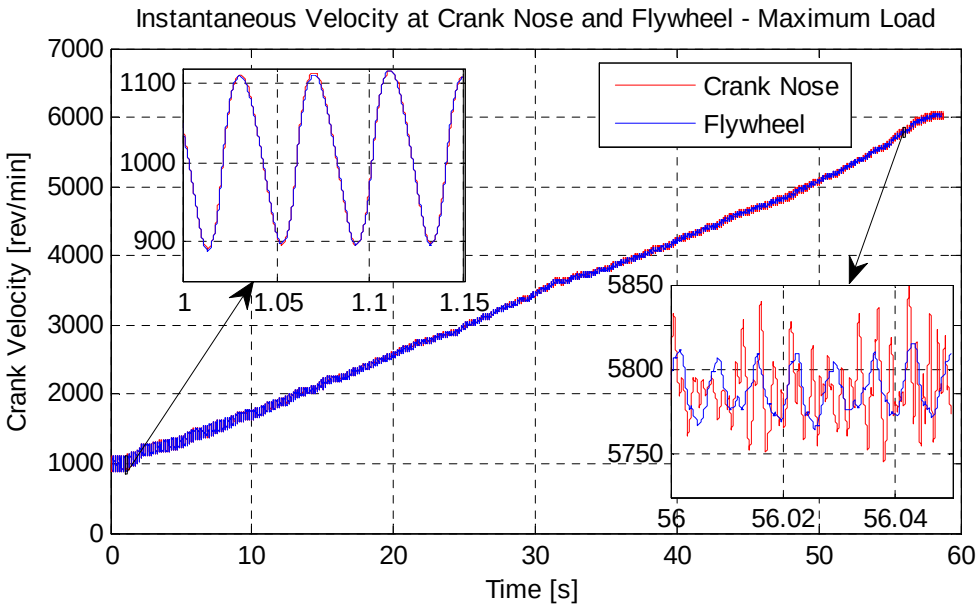


Figure 5.23 – Instantaneous rotational velocity estimates from the crank nose (encoder) and the flywheel (inductive probe using Hilbert de-modulation)

5.7 Crankshaft Torsional Analysis

With rotational velocity information now available with high angular resolution from both ends of the Ford I3 crankshaft – from the crank nose via the calibrated degree marks of the Kistler encoder and from the flywheel via the Hilbert transform demodulated flywheel teeth signal, it is possible to understand how the crankshaft twists across the engine's operating range.

Understanding the flexible kinematics of the crankshaft is likely to be important for cylinder pressure reconstruction, as the relationship between cylinder pressure and measured crank angular acceleration will be different in regimes of rigid and flexible crank dynamics. An ANN successfully trained at an operating condition where the crankshaft is rigid may not generalise successfully at an operating condition where the crankshaft is flexible, as the crank acceleration may have a different relationship to the cylinder pressure sourced excitation.

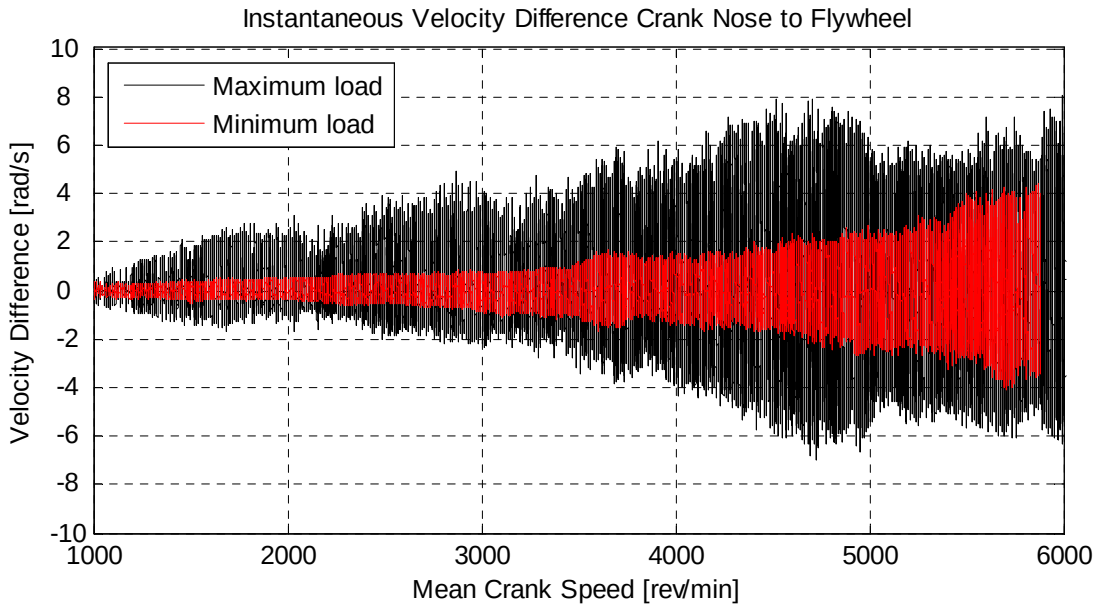


Figure 5.24 – Instantaneous velocity difference between crank nose (measured by encoder) and flywheel (measured by inductive probe) across the engine speed range at maximum (black) and minimum (red) loads

The velocity signals from each end of the crankshaft are subtracted, and the resulting time history is shown in Figure 5.24 – the red curve shows data from a minimum load test, and the black curve data from a maximum load test. The amplitude envelope can be seen to reach a maximum at around 4500 rev/min, suggesting this is the speed at which the crankshaft flexible response is highest. Low speed detail views of the data at each end of the crankshaft are shown in Figure 5.25.

More information can be seen in Figure 5.26 (minimum load) and Figure 5.27 (maximum load), where short time Fourier transforms have been implemented to study the frequency content present in the crank differential velocity across the engine's speed range.

Here it can be clearly seen (particularly so in the maximum load case; Figure 5.27) that each engine order (the diagonal lines) excites a relatively broad resonant response in a frequency range of approximately 320-380 Hz. The 4.5 Order (2^{nd} harmonic of firing frequency) results in the highest amplitude response, equating to a crank speed of approximately 4700 rev/min.

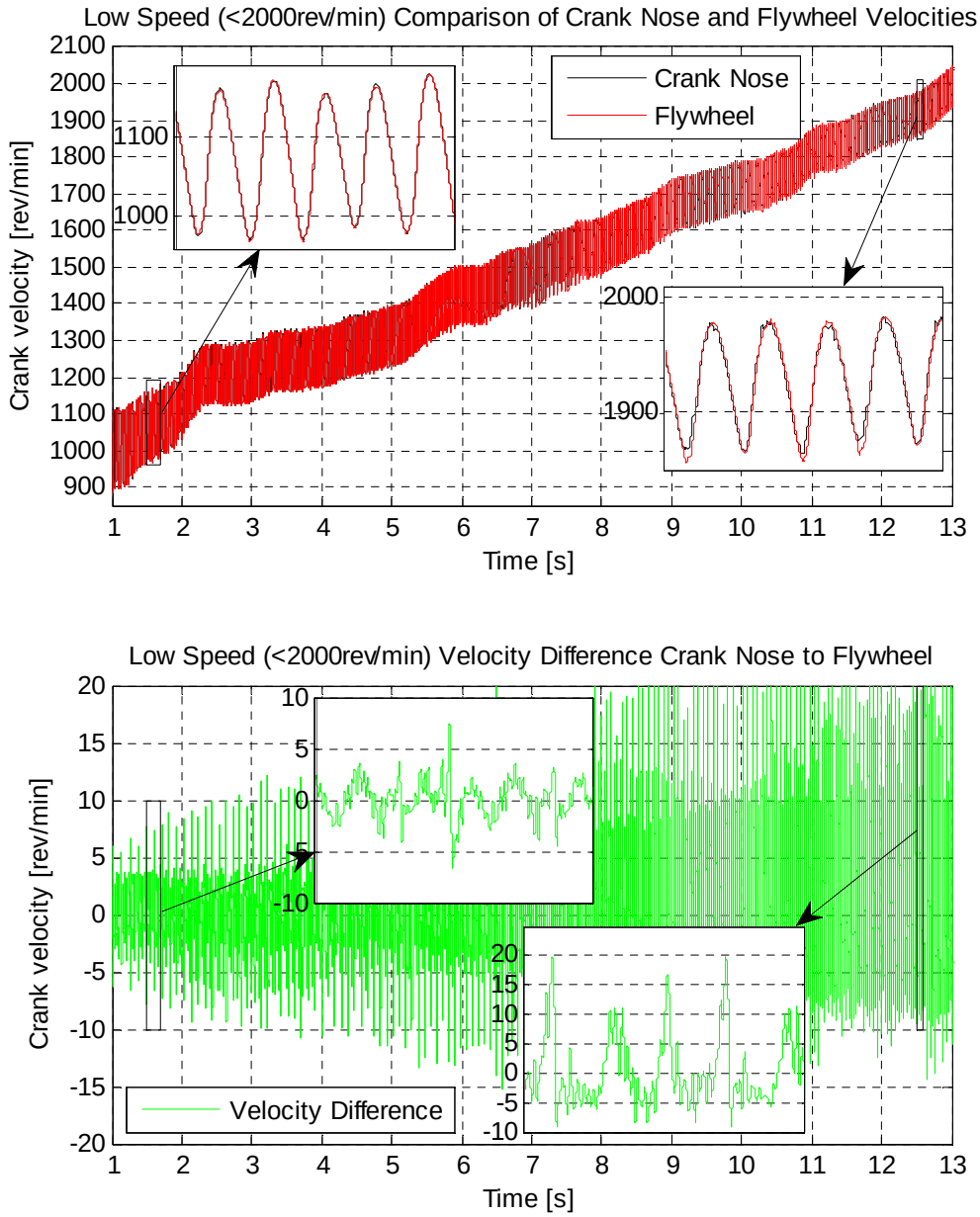


Figure 5.25 – Comparison of crank nose and flywheel velocity estimates and their difference at low speed (<2000 rev/min, maximum load)

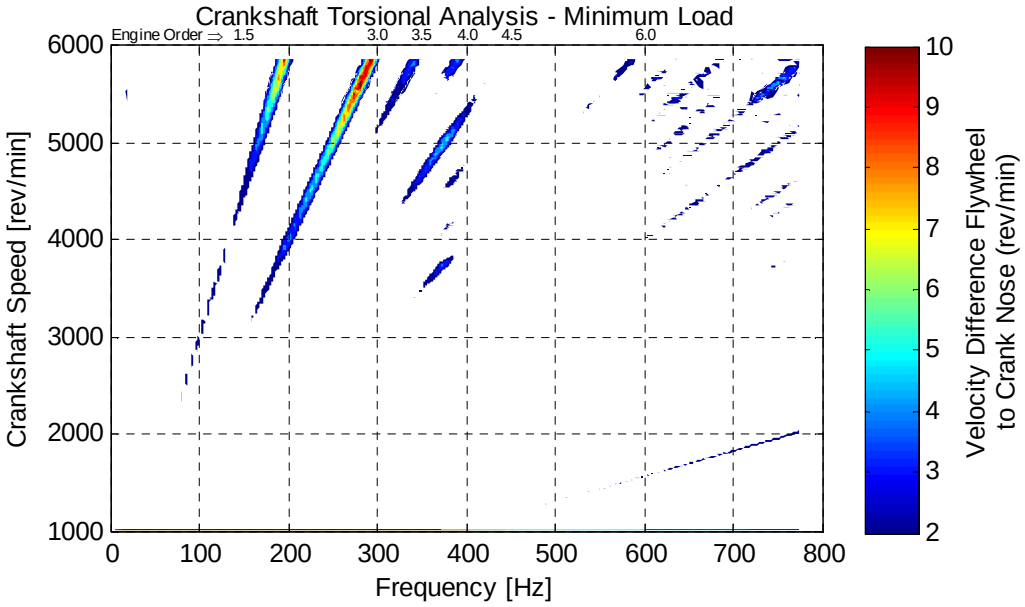


Figure 5.26 – Frequency vs speed map of crankshaft relative angular velocity (cranknose velocity subtracted from flywheel velocity) across the engine speed range 1000-6000 rev/min at minimum load

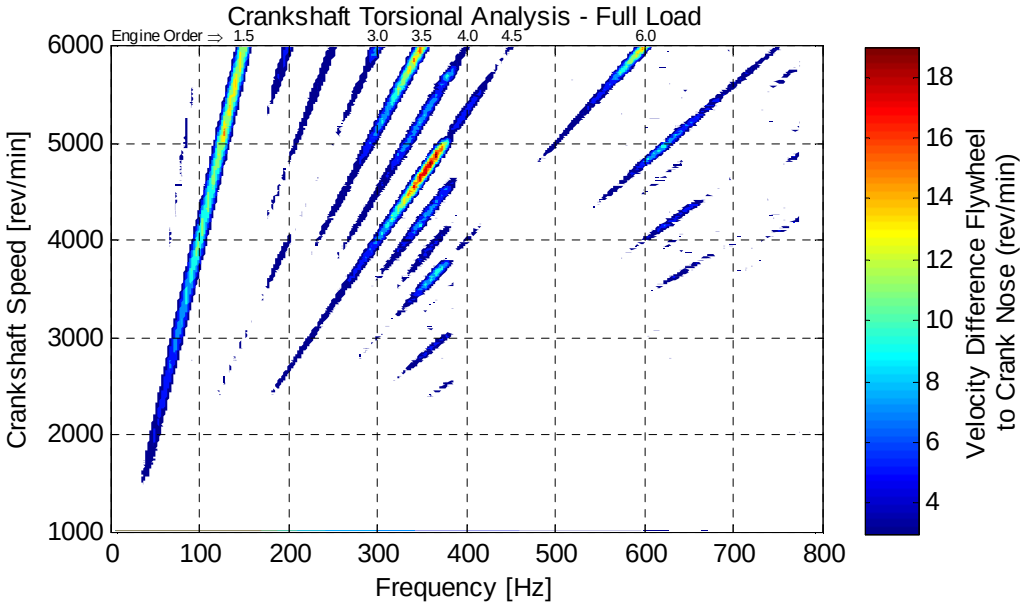


Figure 5.27 – Frequency vs speed map of crankshaft relative angular velocity (cranknose velocity subtracted from flywheel velocity) across the engine speed range 1000-6000 rev/min at full load

5.8 Knock sensor and accelerometer data reduction

Cylinder block vibration provides an alternative route towards cylinder pressure reconstruction. Several researchers have studied this approach

with interesting results [Gao and Randal 1999], [Vulli 2006], [Bizon et al. 2011]. The Ford I3 engine is instrumented with 2 block vibration sensors – an instrumentation quality piezo electric accelerometer (exhaust side) and a production standard knock sensor (intake side), as described in section 4.3.

As the dynamic pressure in the cylinder impinges on the internal faces of the cylinder head, bore and piston crown, the resulting forces transferred into the engine's structure will result in a vibration response, which is measured by the vibration sensors. In this case, the transfer functions encoding the cylinder pressure information are complex, with transmission routes through the structure being different from each cylinder to each of the sensors. The different sensor frequency responses and different locations of the sensors offer several possibilities for extracting cylinder pressure relevant data. Additionally, the cylinder pressure is one of many forces exciting vibration in the structure, among others including valve opening/closing and injector events, piston slap and crank train inertia forces.

This section of the thesis seeks to better understand the vibration signatures available from the transducers and to consider directions for use in ANN training.

Valve Events

The 4-stroke in-line 3 cylinder engine is quite well suited to this study, as the firing and valve timing events are reasonably well separated in the crank angle domain. Table 5.5 and Table 5.6 list the occurrences of valve opening and closing events through the engine cycle.

Valve Event	Crank Angle Location Relative to TDC Firing
Inlet Valve Opening (IVO)	354°
Inlet Valve Closing (IVC)	-138°
Exhaust Valve Opening (EVO)	140°
Exhaust Valve Closing (EVC)	-356°

Table 5.5 - Position of valve events relative to TDC firing (common for each cylinder)

Valve Event	Crank Angle Location Relative to TDC Non-firing Cylinder 1		
	Cylinder 1	Cylinder 2	Cylinder 3
IVO	714°	474°	234°
IVC	222°	702°	462°
EVO	500°	260	20
EVC	4°	484	244°

Table 5.6 – Position of valve events within an engine cycle starting at TDC non-firing cylinder No. 1

Figure 5.28 shows the data from Table 5.5 and Table 5.6 graphically, relating the occurrence of the firing events and the corresponding valve events for each cylinder through the engine cycle.

The figure shows that the important P_{max} timing for each cylinder is well separated from the various opening and closing events that introduce significant force input to the structure and result in vibration response. Other events, particularly piston slap are unavoidably close to the P_{max} timing, and it is for effective network training to be able to discriminate the response to these inputs.

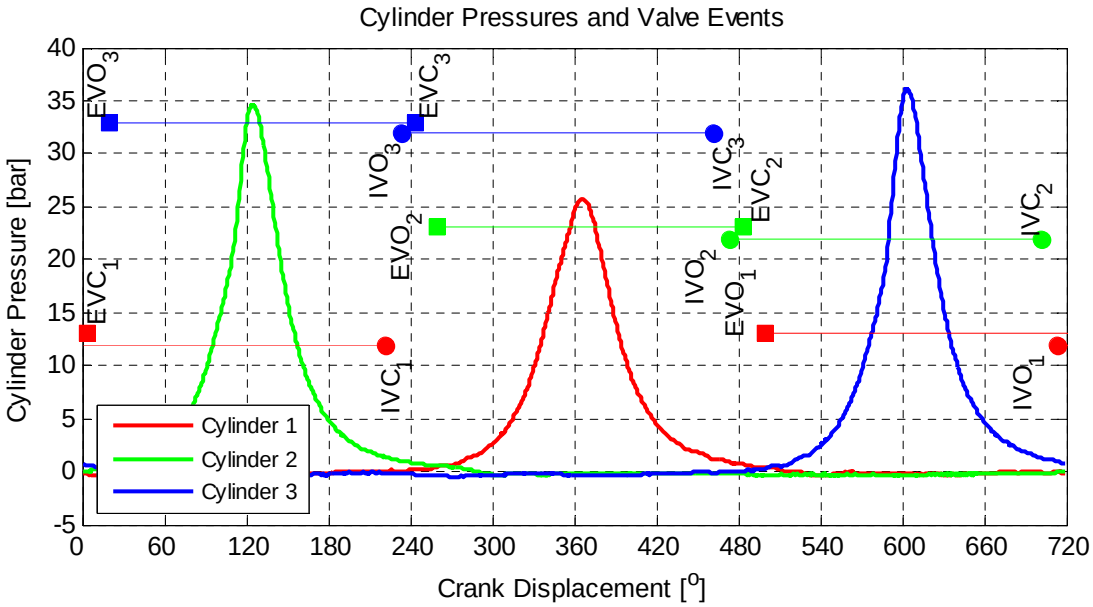


Figure 5.28 – Occurrence of cylinder pressure and valve timing events through the engine cycle (example from 1000 rev/min, 10 Nm dataset)

Vibration Data Frequency Content

The differing force inputs to the engine structure may result in responses at different frequencies at the vibration sensor locations. Selection of appropriate frequency ranges to present to an ANN may provide a better basis for successful training. A challenge to understanding the frequency content of the signals is their non-stationary nature. Their frequency content changes rapidly with time and the use of a basic Fourier analysis would estimate only the average energies for each frequency across the whole cycle.

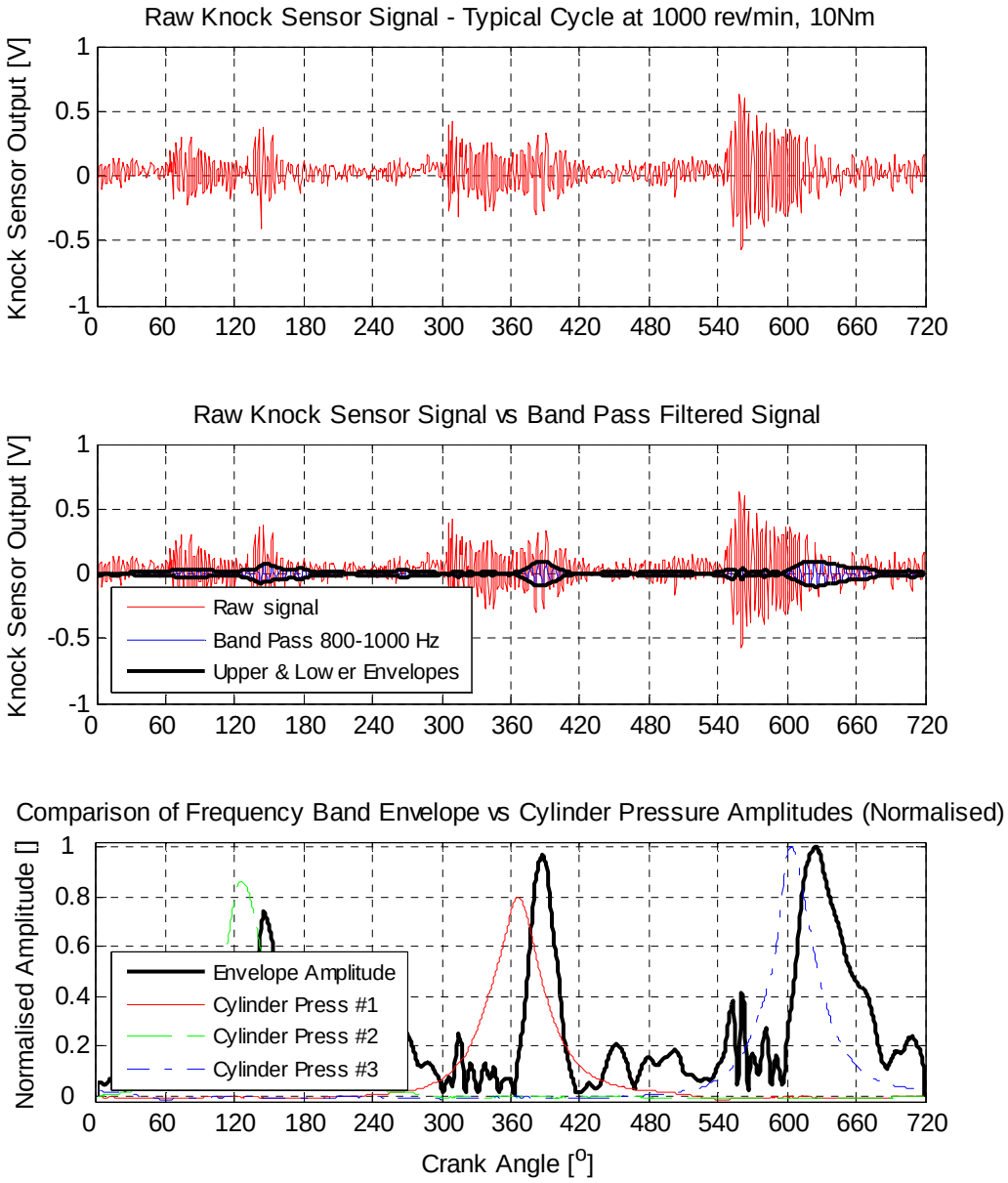


Figure 5.29 – Example of time-frequency analysis of knock sensor signal: 800-1000 Hz at 1000 rev/min, 10 Nm

A number of time-frequency analysis techniques are available to study the time-variant frequency content, including the Short Time Fourier Transform (STFT) [Vulli et al. 2008], Wigner-Ville Distribution [Albarbar et al. 2010] and Wavelet Analysis [Sen et al. 2008] [Wu, Chan 2006]. The method chosen to study the frequency content of the signals in this work utilises a bank of zero phase band pass filters, each 200 Hz wide, the resulting filtered signals are then bounded by upper and lower envelope functions, and the difference between these envelopes gives the instantaneous peak to peak amplitude estimate for that frequency band.

Figure 5.29 shows the raw signal from the knock sensor at 1000 rev/min, 10 Nm load. Three bursts of energy can be seen in the upper graph, but these do not obviously correlate with combustion pressure events, as they appear to generally show more energy before the TDC firing positions at 360°, 120° and 600° for cylinder 1, 2 and 3 respectively. The energy in these high vibration amplitude regions are more coincident with the compression phase of the cylinder pressure trace. Band pass filtering and extracting the envelope amplitude for the frequency range 800 Hz - 1000 Hz (as an example) gives a signal which may correlate better to combustion pressure.

The centre graph shows the band pass filtered time history with the upper and lower envelope functions applied, the lower graph compares the normalised envelope function for 800 Hz - 1000 Hz against normalised cylinder pressure. For this example cycle, the envelope amplitudes show a better correlation to combustion pressures (albeit with some delay), with the peaks timed reasonably consistently, and the relative magnitudes broadly correct.

Extracting envelope amplitudes from each of the band pass filters allows a time-frequency map to be generated through the engine cycle. Figure 5.30 shows the time-frequency content of the knock sensor signal across a typical engine cycle at 1000 rev/min, 10 Nm. The z-axis displays log10 of the amplitude values in order to increase the dynamic range of the colourmap image. Figure 5.33 shows similar data for the piezo-electric accelerometer.

Knock Sensor Time-Frequency Distribution at 1000 rev/min, 10 Nm

The knock sensor data shown in Figure 5.30 shows much less low frequency (<600 Hz) content than seen for the accelerometer data in Figure 5.33. This is expected, as the knock sensor has relatively poor low frequency response.

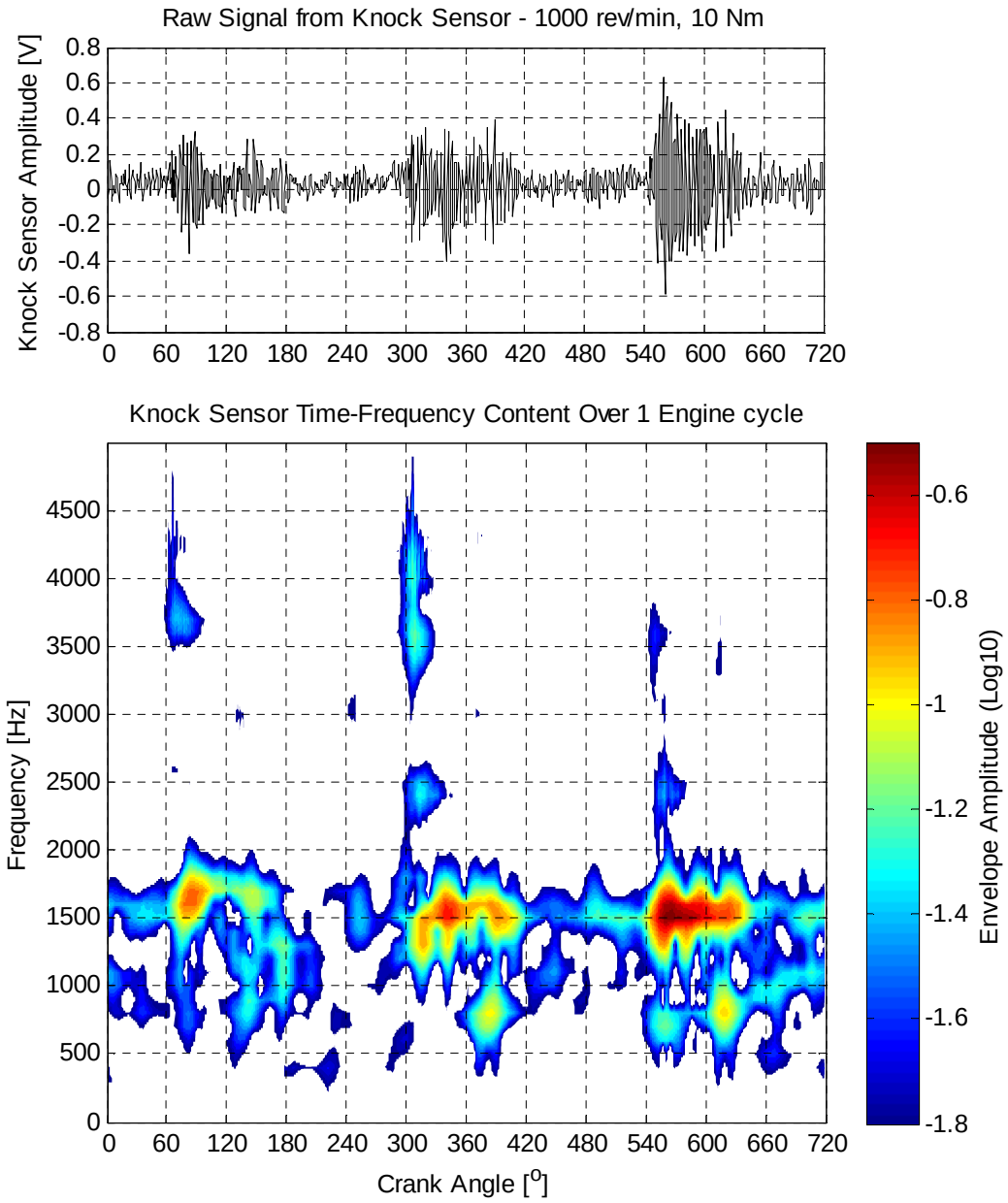


Figure 5.30 – Knock sensor signal: example time history (upper) and time-frequency distribution (lower) over one engine cycle at 1000 rev/min, 10 Nm

Knock sensor data in Figure 5.30 shows 3 distinct regions of frequency content. The most obvious is centred around 1500 Hz and shows 3 distinct energy releases, each centred around the TDC firing crank positions of 120°,

360° and 600°, however this frequency does tend to show high amplitude well before the peak cylinder pressure occurs and can show a local reduction in amplitude around TDC – possibly suggesting the excitation is more related to rate of pressure change or crank acceleration rather than pressure magnitude. The second frequency potentially related to cylinder pressure magnitude is centred around 800 Hz, and tends to occur 10° - 20° after TDC firing.

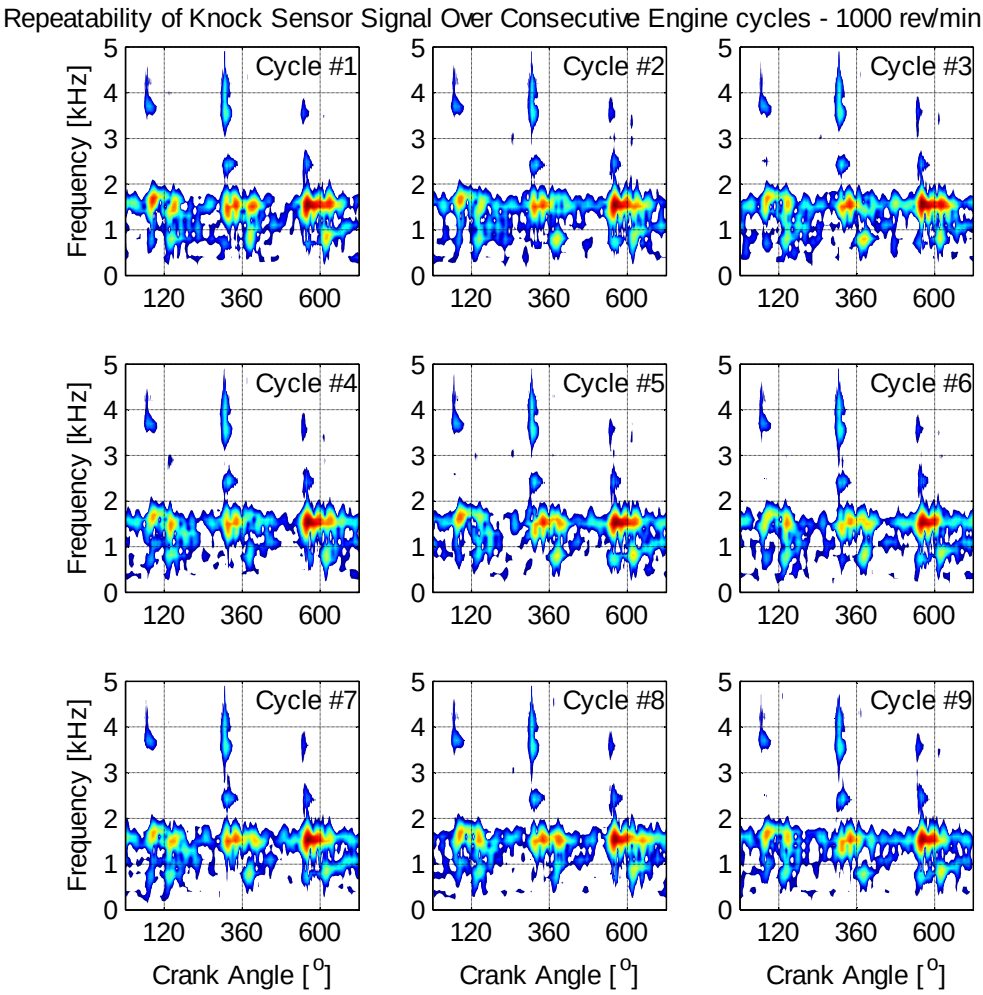


Figure 5.31 – Repeatability of knock sensor time-frequency distribution over 9 consecutive engine cycles at 1000 rev/min, 10 Nm (log10 amplitude, fixed scale for all conditions)

The third frequency range of note contains the broad band energy releases seen between 2000 Hz-4500 Hz, typically occurring ≈60° before TDC firing. Such a broad band response is typical of an impulsive excitation, and it seems unlikely that this is related directly to pressure magnitude but may be

a mechanical noise source. As a result, frequencies above 2000 Hz will be excluded from cylinder pressure reconstruction efforts.

Figure 5.31 shows similar time-frequency distributions for nine consecutive engine cycles operating at 1000 rev/min and 10 Nm load. The distributions appear quite repeatable cycle to cycle. Figure 5.32 shows time-frequency distribution for a single engine cycle for all nine recorded steady state operating conditions. The frequency ranges discussed above for 1000 rev/min, 10 Nm load appear to be relevant to the other operating conditions as well.

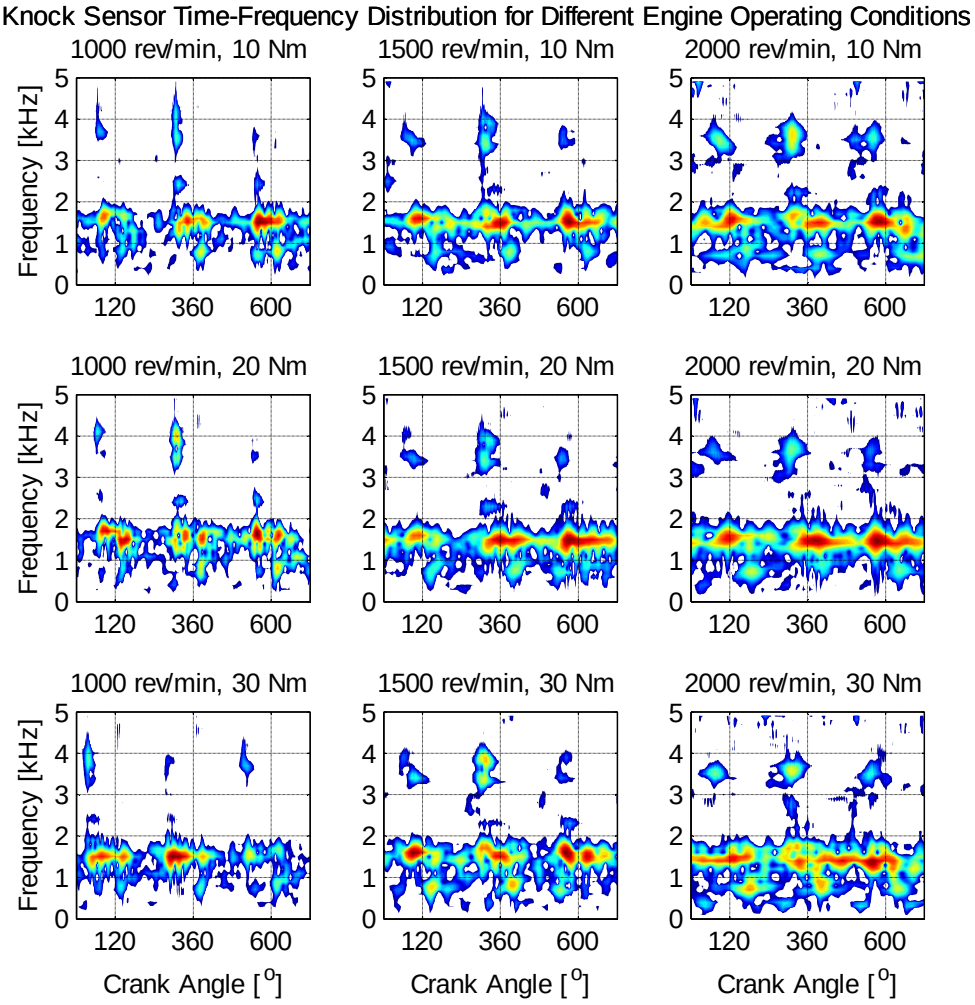


Figure 5.32 – Knock sensor time-frequency distribution over one engine cycle at different engine operating conditions (log10 amplitude, fixed scale for all conditions)

Accelerometer Time-Frequency Distribution at 1000 rev/min, 10 Nm

The accelerometer data shown in Figure 5.33 has a number of similarities to the knock sensor data discussed above.

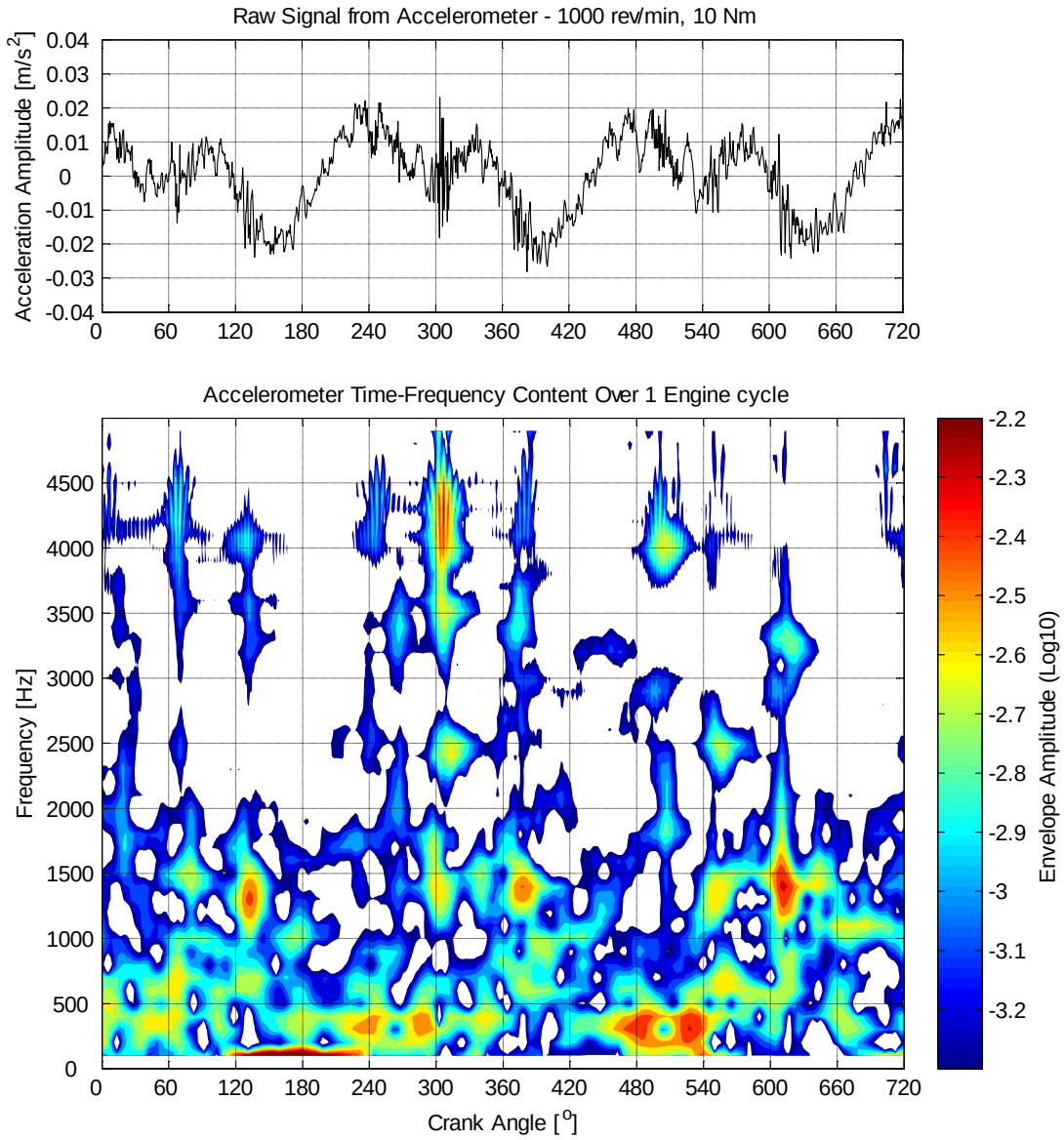


Figure 5.33 – Accelerometer signal: example time history (upper) and time-frequency distribution (lower) over one engine cycle at 1000 rev/min, 10 Nm

A similar pattern of frequencies around between 1200 Hz-1500 Hz is present around TDC firing as was seen centred on 1500 Hz for the knock sensor, but the pattern between 200 Hz -1000 Hz is less clear with a broader distribution across the cycle. High frequency 2000 Hz -5000 Hz, broad band events are also seen in the accelerometer data, but more events are visible, in particular

an impulsive event can be seen occurring shortly after TDC firing that was not present in the knock sensor data – this is likely to be piston slap, visible on the accelerometer as this is on the exhaust side of the engine which is the thrust side of cylinder bore where piston slap occurs. The upper graph in Figure 5.33 clearly shows very low frequency content from the accelerometer with a 3 per cycle (i.e. 1.5 engine order – firing frequency) repetition, suggesting this content is torque recoil excited roll of the engine on its mounting system, driven by the reaction to the same forces responsible for crankshaft acceleration. As the accelerometer is positioned horizontally on the side of the cylinder block, it is quite well positioned to capture the engine roll action due to torque recoil.

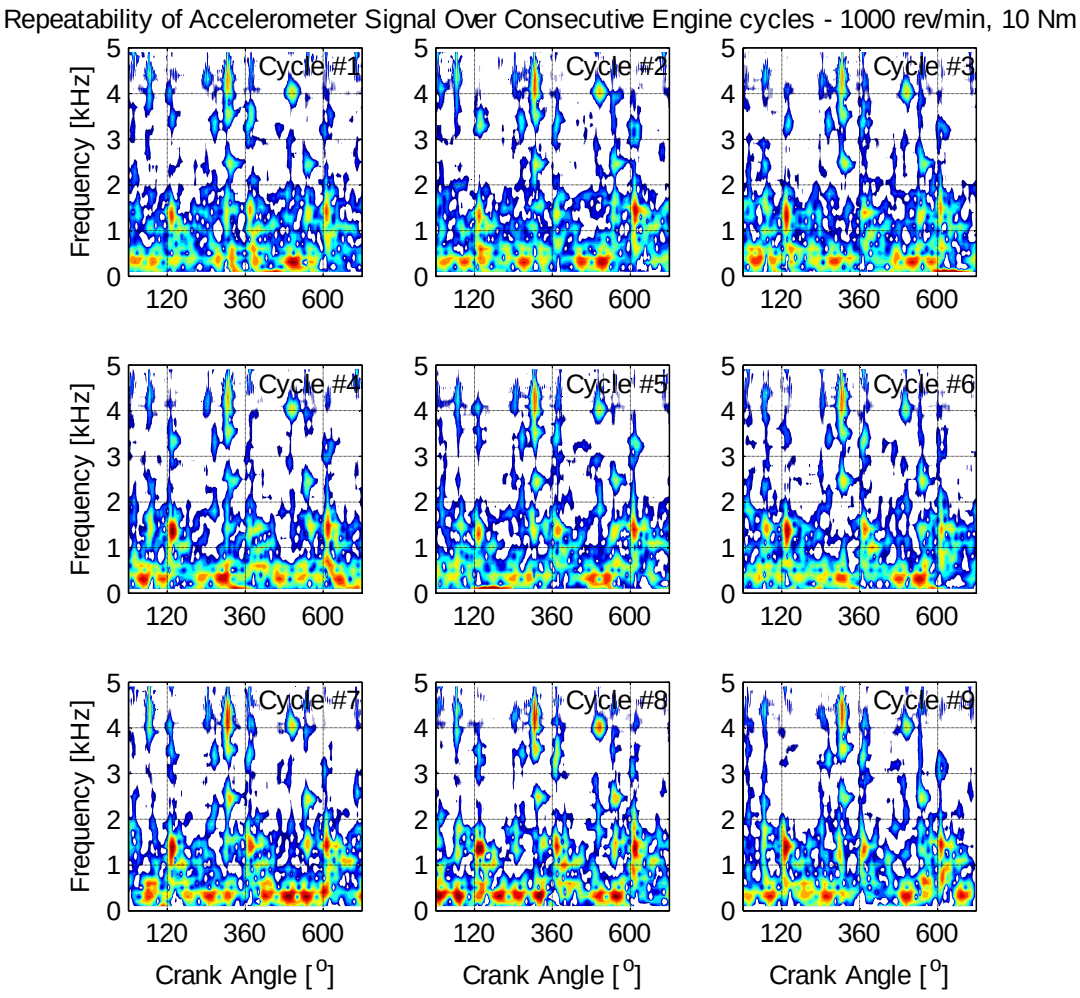


Figure 5.34 – Repeatability of accelerometer time-frequency distribution over 9 consecutive engine cycles at 1000 rev/min, 10 Nm (log10 amplitude, fixed scale for all conditions)

Figure 5.34 shows the time-frequency distribution for 9 consecutive engine cycles at 1000 rev/min, 10 Nm load, and again shows a quite repeatable pattern.

Figure 5.35 shows data for a single cycle from each of the nine steady state operating conditions recorded – the frequencies and timing of interest appear to be broadly consistent as the conditions change, in particular the low frequency components (<500 Hz) increase in amplitude as the torque recoil excitation increases with increasing brake load, and the 1200 Hz -1500 Hz band shows some amplitude sensitivity with increasing speed and load.

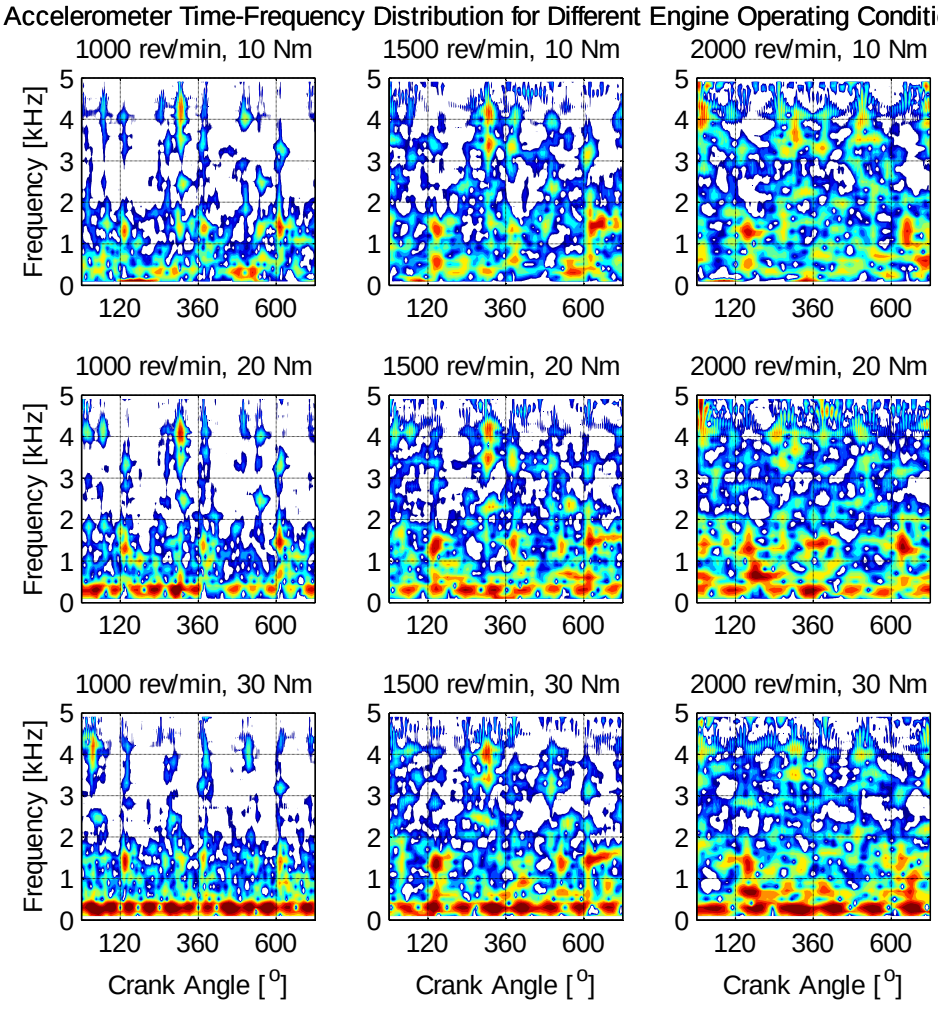


Figure 5.35 – Accelerometer time-frequency distribution over one engine cycle at different engine operating conditions (log10 amplitude, fixed scale for all conditions)

Vibration Signal Coherence with Cylinder Pressure Signals

A significant difficulty associated with using the block vibration signals (accelerometer or knock sensor) is rooted in the large number of force inputs to the engine's structure, all of which will result in a vibration response of some magnitude. Many of these inputs occur at similar crank shaft angle timings as cylinder pressure inputs. It is useful to understand the relative contribution of the cylinder pressure sourced inputs to the measured vibration responses.

The frequency domain coherence function described in equation (5.3) is useful to understand the linearity existing between input and output signals, and is commonly used to understand the degree to which the output measurement depends on a given input.

$$\gamma^2 = \frac{\left|G_{xy}\right|^2}{G_{xx} \bullet G_{yy}} \tag{5.3}$$

Where:
γ^2 = magnitude squared coherence
G_{xy} = Frequency domain cross-power between input and output (averaged)
G_{xx} = Frequency domain auto-power of input signal (averaged)
G_{xx} = Frequency domain auto-power of input signal (averaged)

To calculate the coherence function, first a relevant window of data was extracted from the time histories of cylinder pressure, accelerometer and knock sensor. The time windows were constructed in a similar manner as those used to understand cylinder pressure frequency content as described in Section 5.2. Figure 5.36 shows an example for the test condition 2000 rev/min, 30 Nm.

The cylinder pressure window around TDC firing is highlighted in the upper graph - from 85° BTDC to 140° ATDC. A corresponding time window is taken from the knock sensor signal as shown in the lower graph, a similar window is also extracted from the accelerometer signal.

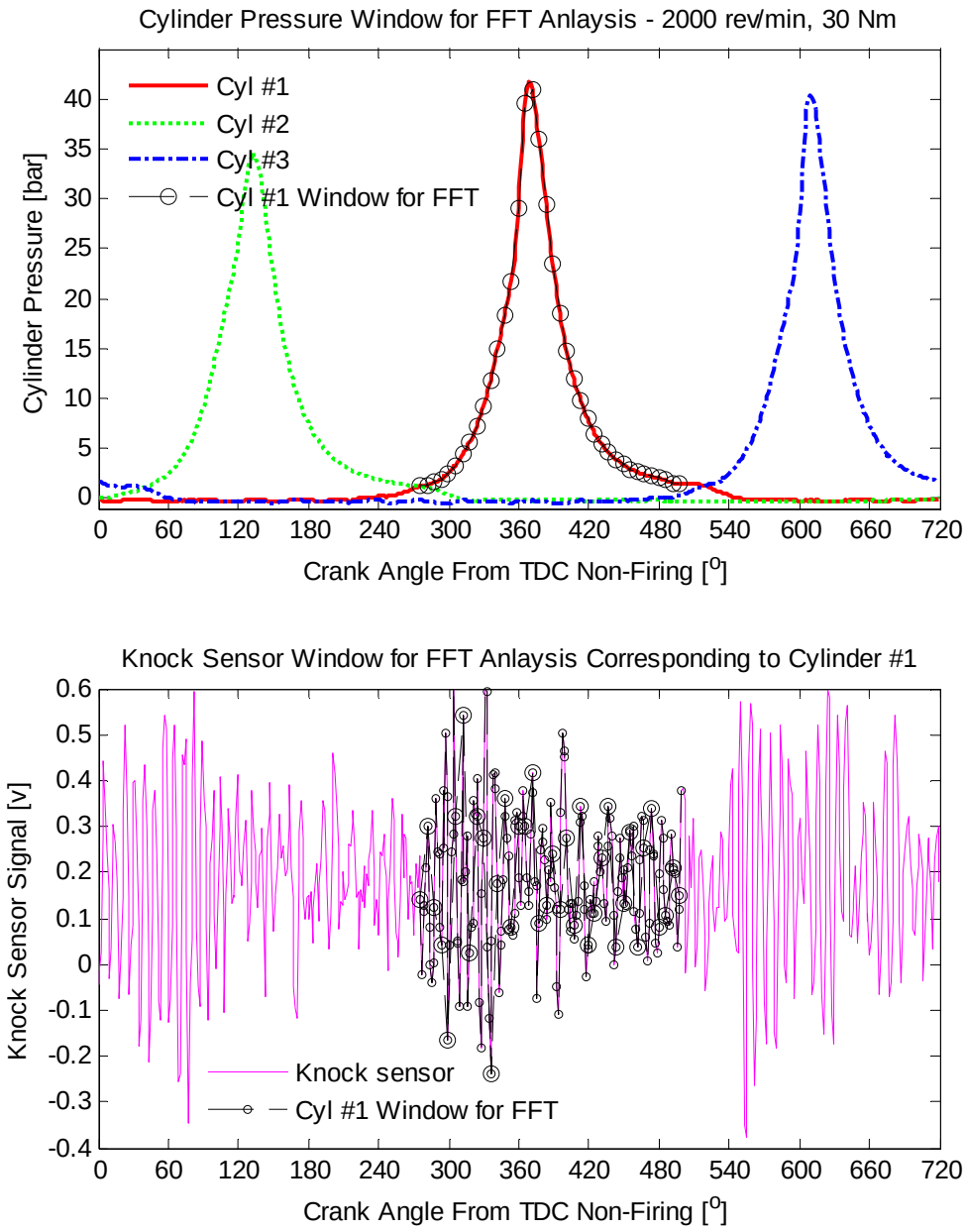


Figure 5.36 – Time window extracted from signals for FFT analysis

Similar steps are taken for each cycle in the data set (in this case, 198 engine cycles), and also for each of the three cylinders with appropriate time windows extracted around their local TDC firing crank angles.

The resulting complex spectra are then used to calculate the magnitude squared coherence function described by equation (5.3). The upper term includes the average cross-power spectrum, and the lower term the average auto-power spectra for input and response signals. The average cross-power will include only energy coherent between the signals, the auto-power will include all energy in the signal. The coherence measure will be in the

range $0 \leq \gamma^2 \leq 1$ with 1 representing the response fully coherent with the input (i.e. all output energy sourced from the input), and 0 meaning no coherence between output and input (i.e. none of output energy sourced from the input).

Figure 5.37 and Figure 5.39 show examples of the coherence function for data from the same 2000 rev/min, 30 Nm operating condition as shown for the crank angle window extraction in Figure 5.36. Figure 5.37 shows the coherence of the accelerometer signal with the pressure signal from each of the three cylinders; Figure 5.39 shows similar data for the knock sensor signal.

It can be clearly seen that above 1000 Hz, there is little coherent energy between the cylinder pressure and either of the response signals - the function has $\gamma^2 \leq 0.25$ above this frequency. This is entirely consistent with the conclusion from section 5.2 which showed the cylinder pressure has little frequency content above 1000 Hz – with very low input energy, there can be little coherent output energy. Coherence is strong $\gamma^2 \geq 0.75$ only below approximately 200 Hz. Generally the coherence for a response from each cylinder's pressure signal is similar.

The coherence function for each cylinder pressure input at each test condition is summarised in Figure 5.38 for the accelerometer, and in Figure 5.40 for the knock sensor, over the reduced frequency range of 0 Hz - 2000 Hz. The conclusion from the example above remains broadly true, with coherence consistent strongest at very low frequencies (< 200 Hz). Some test condition and cylinder combinations show some stronger coherence up to 500 Hz.

This data suggests that the best engine structural vibration frequencies to present to a cylinder pressure reconstruction method may be 0 Hz – 200 Hz or possibly 0 Hz - 500 Hz.

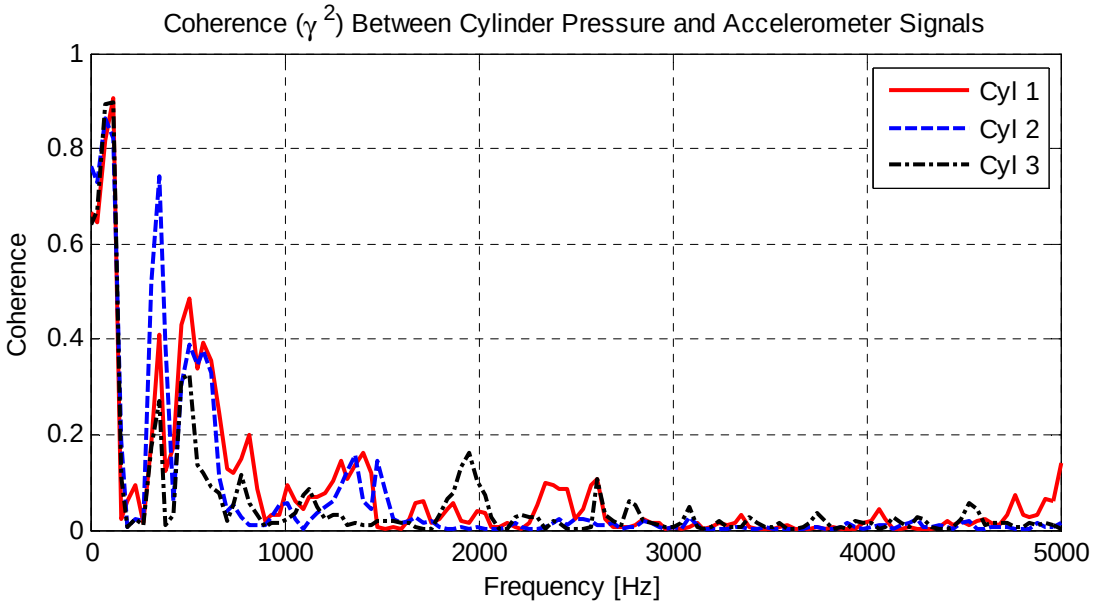


Figure 5.37 – γ^2 between cylinder pressure and cylinder block accelerometer signals over 198 cycles, 2000 rev/min, 30 Nm

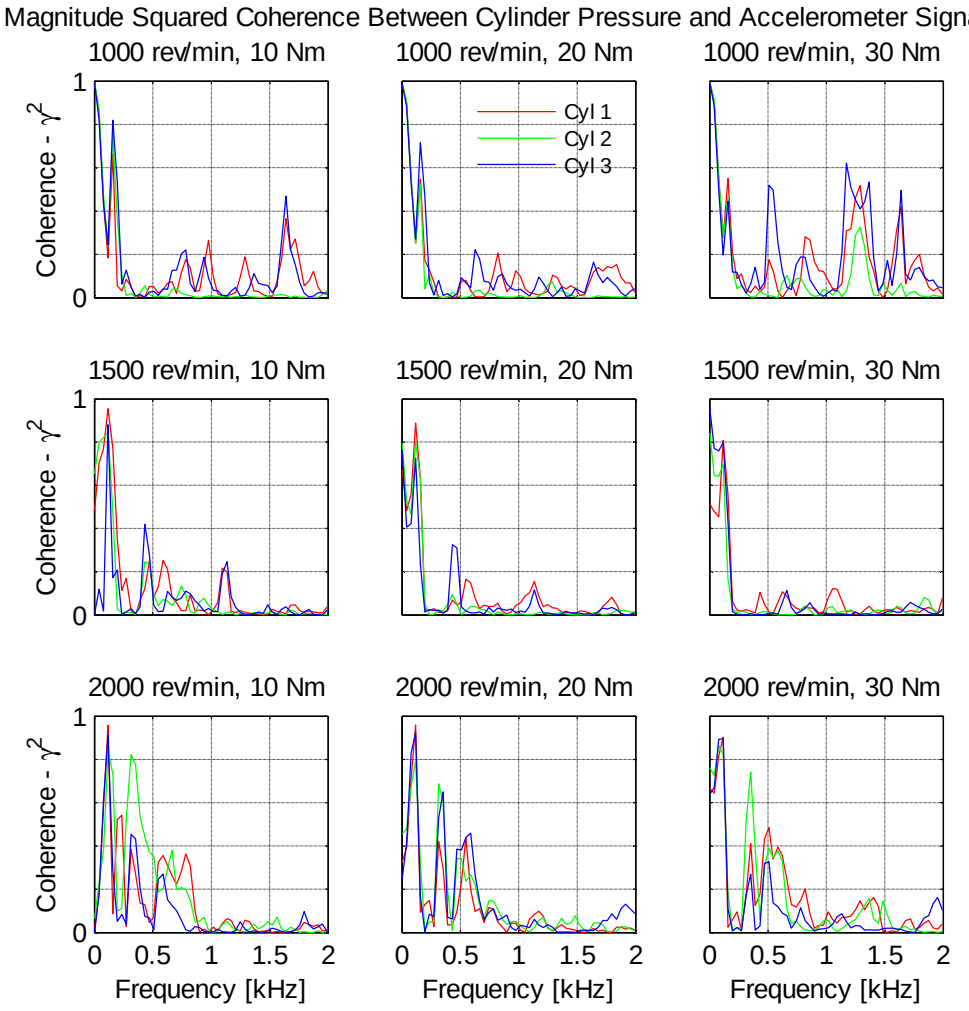


Figure 5.38 – Cylinder pressure to accelerometer coherence function for all test conditions

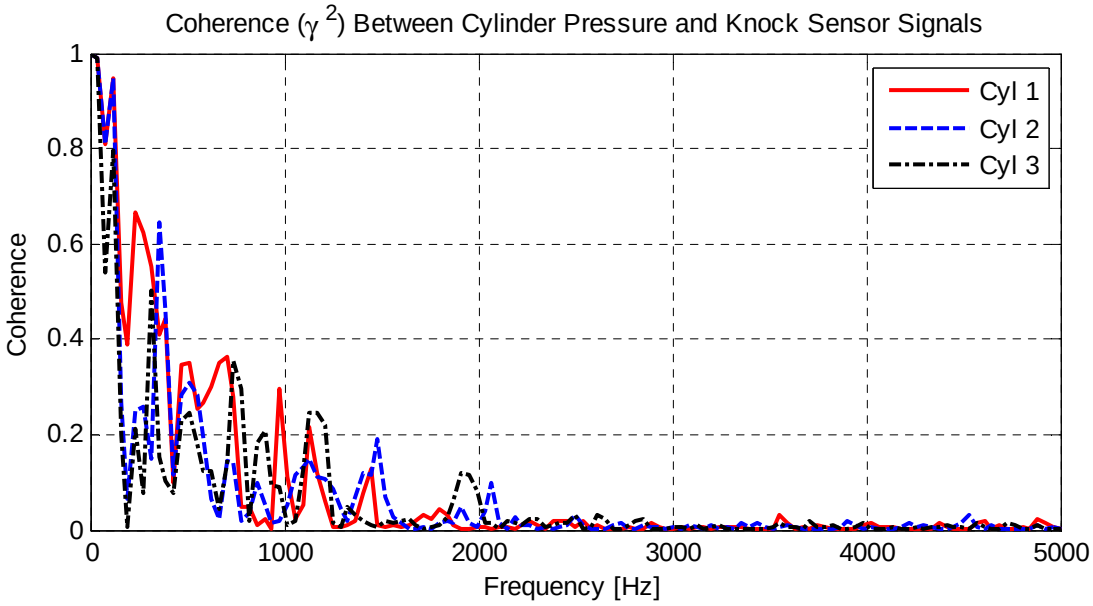


Figure 5.39– γ^2 between cylinder pressure and knock sensor signals over 198 cycles, 2000 rev/min, 30 Nm

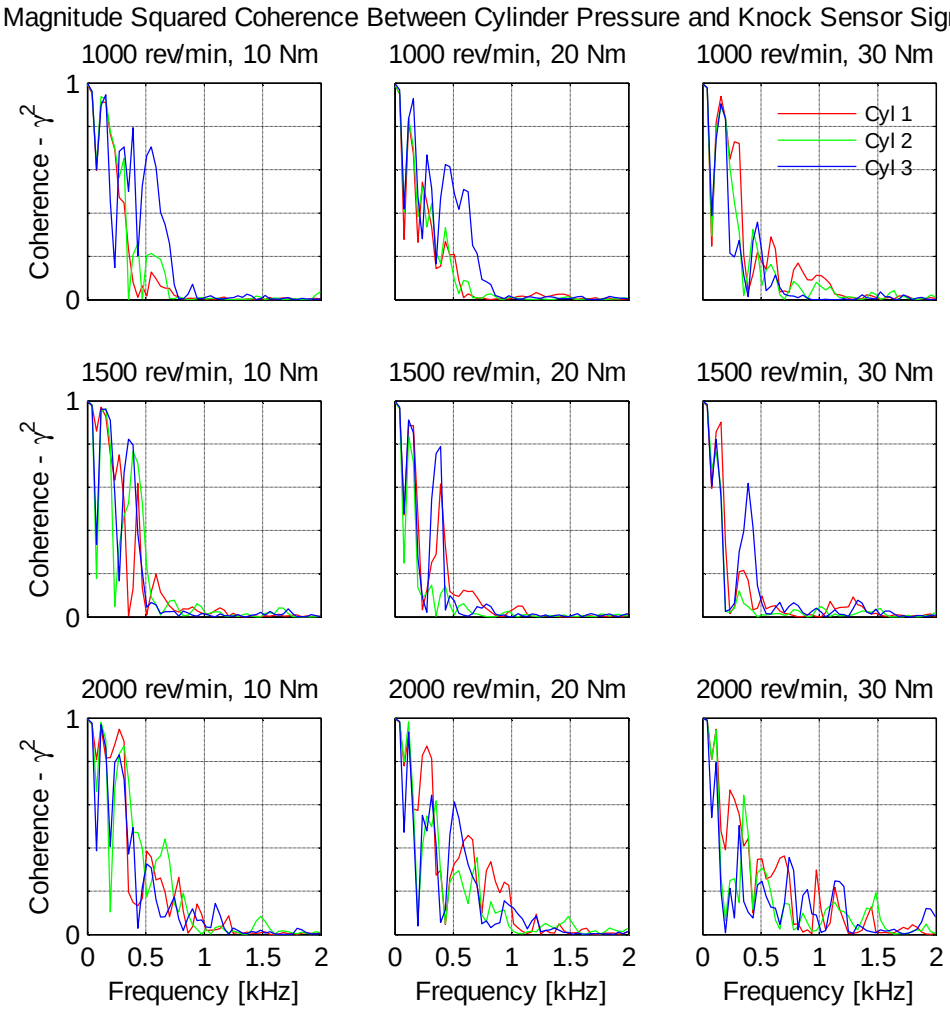


Figure 5.40 – Cylinder pressure to accelerometer coherence function for all test conditions

Chapter Six

6. CYLINDER PRESSURE RECONSTRUCTION VIA ARTIFICIAL NEURAL NETWORKs

This chapter documents results from training networks in a feed forward architecture. These trials consist of:

- Pure feed forward architecture using angle domain input vectors
- Pure feed forward architecture using time domain input vectors
- NARX recurrent architecture trained in a feed forward manner using teacher forcing, and tested in both feedforward and true recurrent connected configuration

All training is conducted in the MATLAB environment using built-in functions to execute the standard back-propagation algorithm. Various input vectors and time delay sets are presented.

6.1 Metrics to Describe the Success of Cylinder Pressure Reconstruction

An important consideration in developing an ANN for cylinder pressure reconstruction is an understanding of what represents a successful result. Time domain predictions from ANNs typically use the square of the error between target and prediction, and this is appropriate where every element of the target vector is equally important.

However, a cylinder pressure signal has some regions that are much more important than others, and performance metrics that focus on the regions with the highest importance would be beneficial. In particular, cylinder pressure around the peak magnitude produced by the firing event on each cylinder is of the greatest interest.

Relevant metrics are particularly useful where large numbers of cycles and / or network training iterations are being studied, when it becomes difficult to

visually evaluate the time domain prediction to target comparison, simply because there is too great a volume to accurately assess in this way.

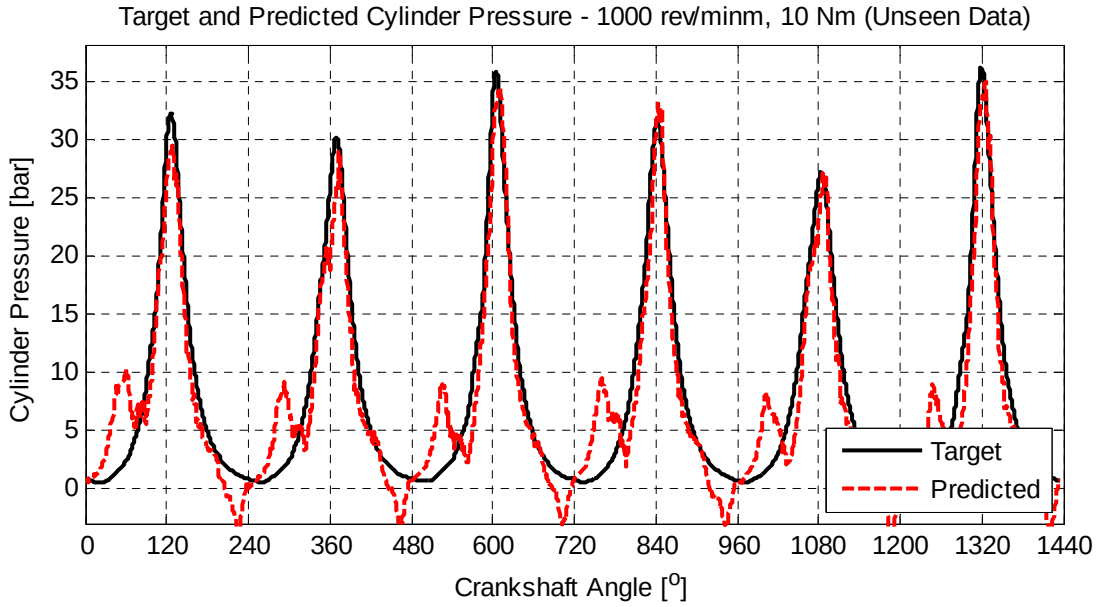


Figure 6.1 – Example target and predicted cylinder pressure using ANN; black – target pressure; red – predicted pressure

Figure 6.1 shows an example of cylinder pressure predicted by an ANN, compared against the target. With two engine cycles of data, the predictions can immediately be judged to be reasonably accurate, with the overall shape of the trace following the target pattern, particularly around TDC firing.

Figure 6.2 presents a squared error metric for the performance of the ANN results shown in Figure 6.1, with the metric simply being $(prediction - target)^2$. It is difficult from this measure to understand how well the network has really performed – the important characteristics are not well presented. In particular the prediction errors away from TDC firing tend to dominate the picture, and at the first level of network performance these are of secondary importance.

Three additional metrics of network performance are used to evaluate results within this thesis:

- Percentage error in the magnitude of peak pressure
- Error in the crank angle position of peak pressure relative to TDC firing
- Mean squared error over a window 30° BTDC to 60° ATDC

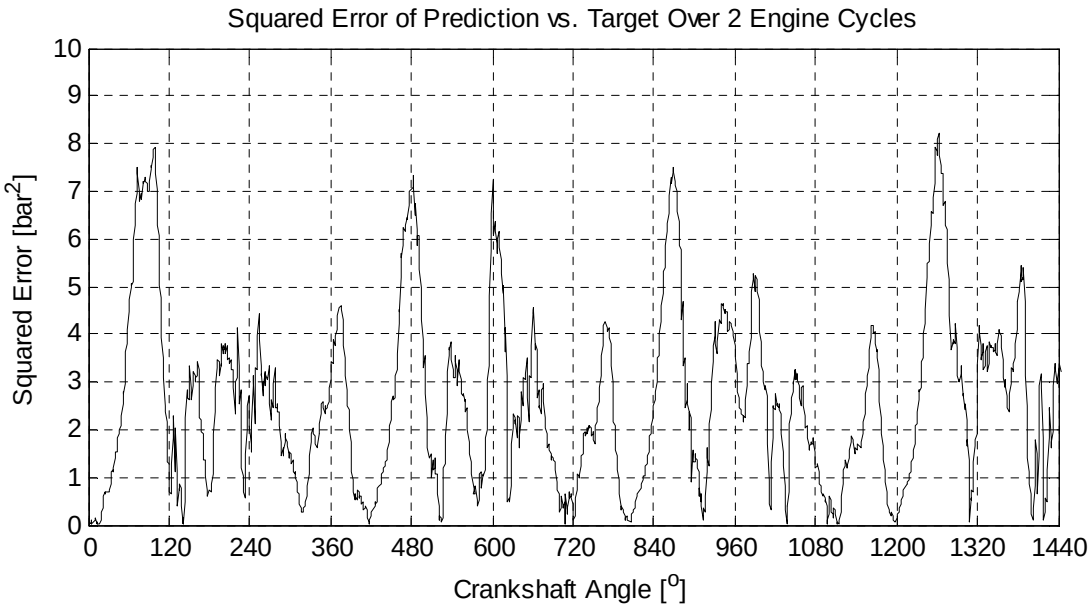


Figure 6.2 – Squared error measure of network prediction performance over 2 engine cycles

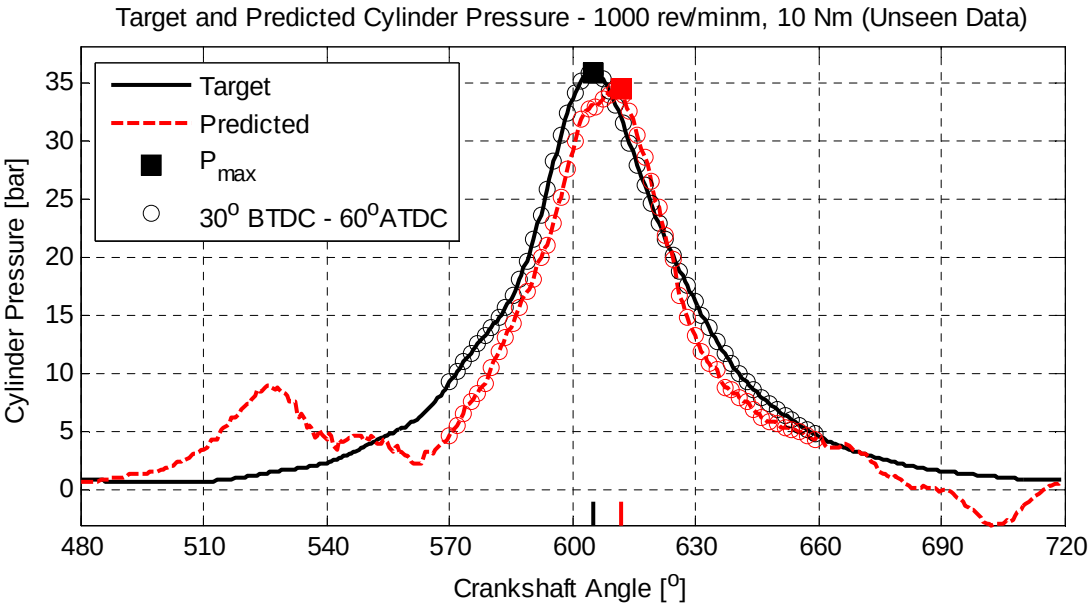


Figure 6.3 – Example of ANN performance metric data extracted from target and predicted cylinder pressures

Figure 6.3 highlights the data points on target and predicted cylinder pressure curves that contribute to these metrics. When network predictions are conducted over many cycles, these metrics can be plotted against cycle number and by cylinder if required. Figure 6.8 shows an example of such a summary plot.

A prediction over a full set of cycles from one dataset will provide approximately 195 cycles to compare target vs. predicted, and hence a total of approximately 585 estimates of each firing pressure metric are available. With these volumes of metrics, summary statistics are calculated consisting of the mean errors and two standard deviations of the values – providing the error range that approximately 95% of the metrics will lie within. Table 6.1 shows summary statistics of P_{max} and θ_{max} errors for the data plotted in Figure 6.8.

6.2 Cylinder Pressure Reconstruction with Feedforward Networks Trained with the Back-Propagation Algorithm

Initially it is useful to demonstrate the limitations of a feed forward network with no recurrent feedback of the predicted values – this sets a benchmark for performance of NARX architectures demonstrated later.

6.2.1 Reconstruction Using Data In The Crank Angle Domain

The preferred method of reconstructing cylinder pressure would be in the time domain. However, some researches [Bizon et al 2011] [Gu et al,1999] have shown interesting results with networks based around crank angle domain data.

To investigate this approach using the reference data set from the Ford I3 engine, a set of inputs and target data were prepared. In each case, a window of data was extracted from the datasets centred around top dead centre firing for each cylinder. Figure 6.4 show an example of the data extracted. The upper graph shows a typical cycle of crankshaft acceleration data at 2000 rev/min, 20 Nm. The dashed lines shows the full cycle of data, the solid lines the segments of data centred around TDC firing for each cylinder.

In each case, the segments extend from 60° BTDC to 60° ATDC – a window of 120°. Prior to extraction of these segments, the data is resampled from the time domain (as it is held in the reference datasets) to the angle domain, with a resolution of 0.5°. Hence each segment of data consists of 241

samples. A segment is extracted for the input (either crank kinematics or knock sensor output etc.) and the target i.e. cylinder pressure.

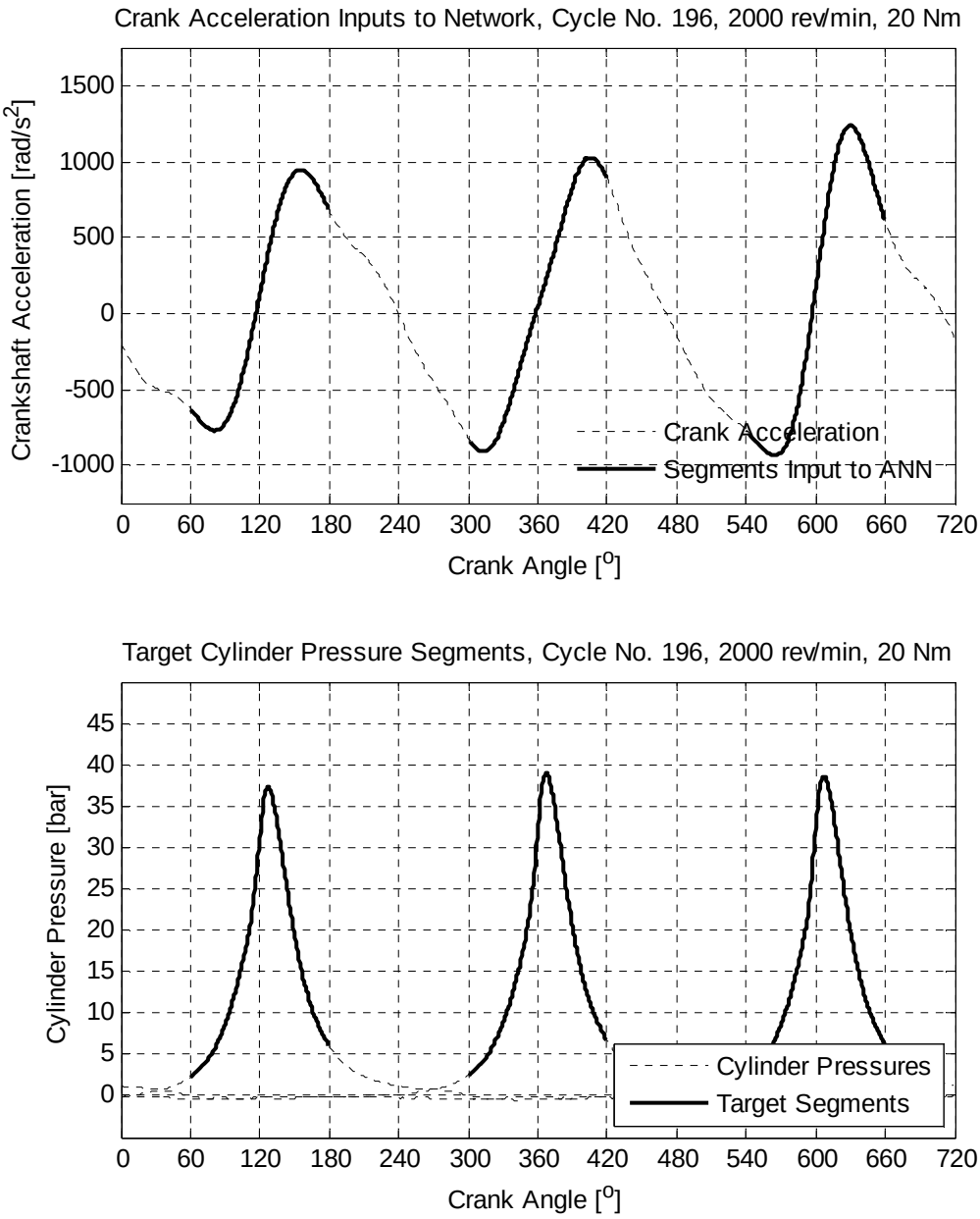


Figure 6.4 – Example data segments used as inputs (upper) and targets (lower) for crank angle domain, feed forward network

A set of windows is extracted from the data for each cycle in the dataset; in this case 198 cycles are constructed. The 198 cycles of data are then presented to a feedforward network consisting of 241 inputs, 10 hidden

neurons and 241 outputs as shown diagrammatically in Figure 6.5 for a network using crank acceleration as input.

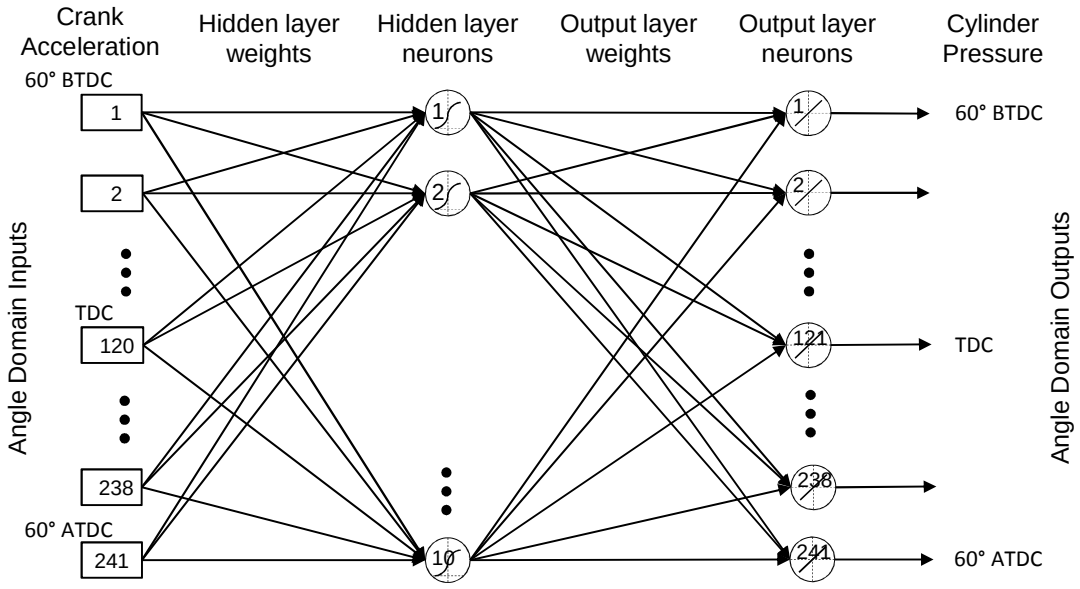


Figure 6.5 – Feedforward network structure for crank angle domain training

The network was constructed and trained using standard MATLAB toolbox functions, with sigmoid activation functions in the hidden layer, and linear functions in the output neurons.

Once trained, the network was then presented with similar input segments extracted from the second reference dataset of the same test condition, once more resampled to the angle domain, and this unseen data used to test cylinder pressure reconstruction.

Initial experiments trained a network using cyclic segments of data centred around TDC firing from cylinder number 1 only. Upon reconstructing pressures from the unseen dataset, the trained network was found to perform well on the data from cylinder number 1, but poorly for the data from cylinder numbers 2 and 3. To attempt to resolve this problem, 3 individual networks were trained each with data segments corresponding to the different cylinders. All networks were of the same structure with 241 inputs, 10 hidden neurons and 241 outputs.

On reconstructing unseen data, each cylinder's data segments were applied to the relevant network. Examples of the reconstruction are shown in Figure 6.6 for networks trained on crank acceleration data and in Figure 6.7 for networks trained to reconstruct from knock sensor data.

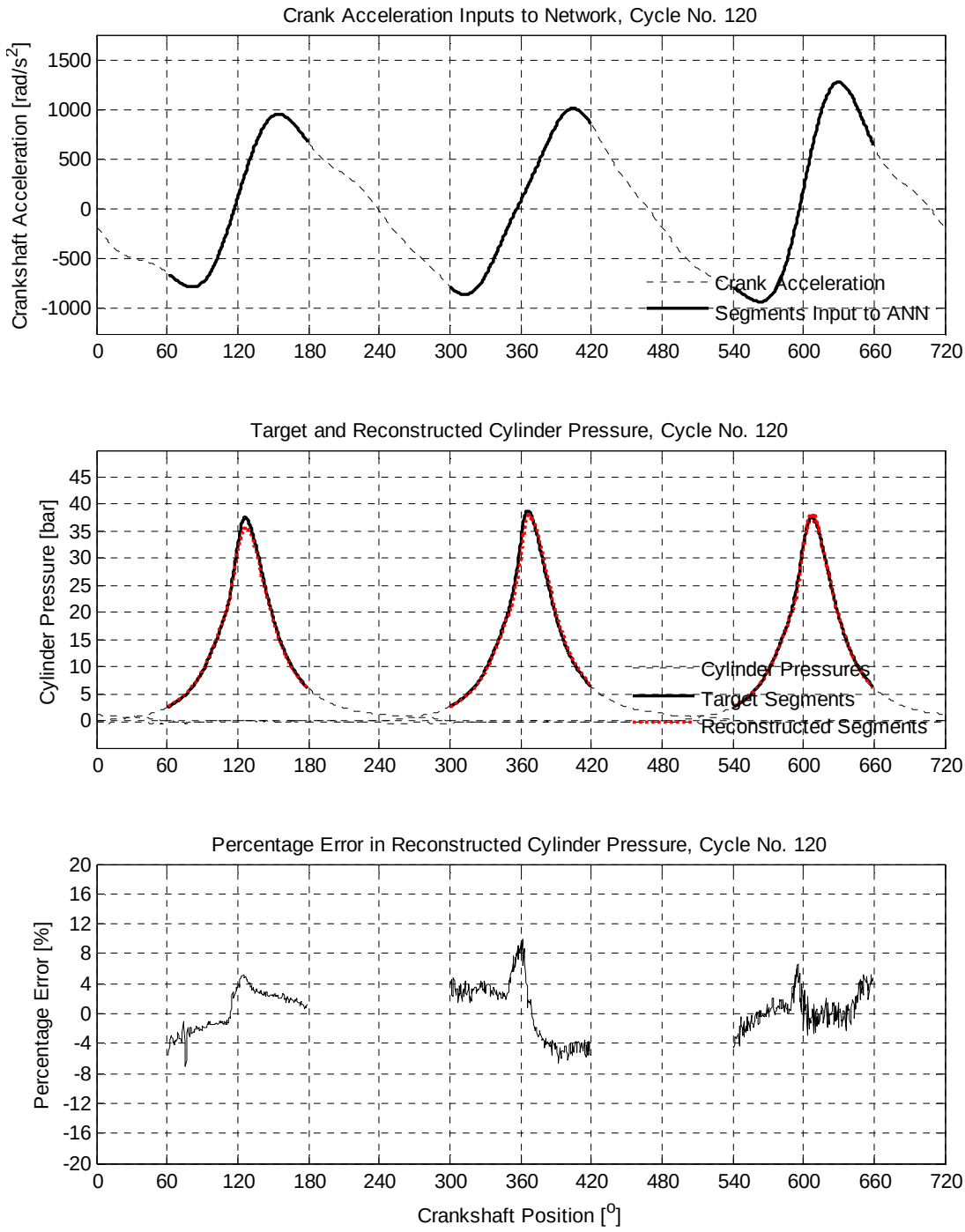


Figure 6.6 – Example of predicted data using unseen crank angle domain crankshaft acceleration data as input – 2000 rev/min, 20 Nm
Upper: Crank acceleration segments used as network inputs
Centre: Target and predicted cylinder pressure segments
Lower: Percentage pressure errors in each reconstructed segment

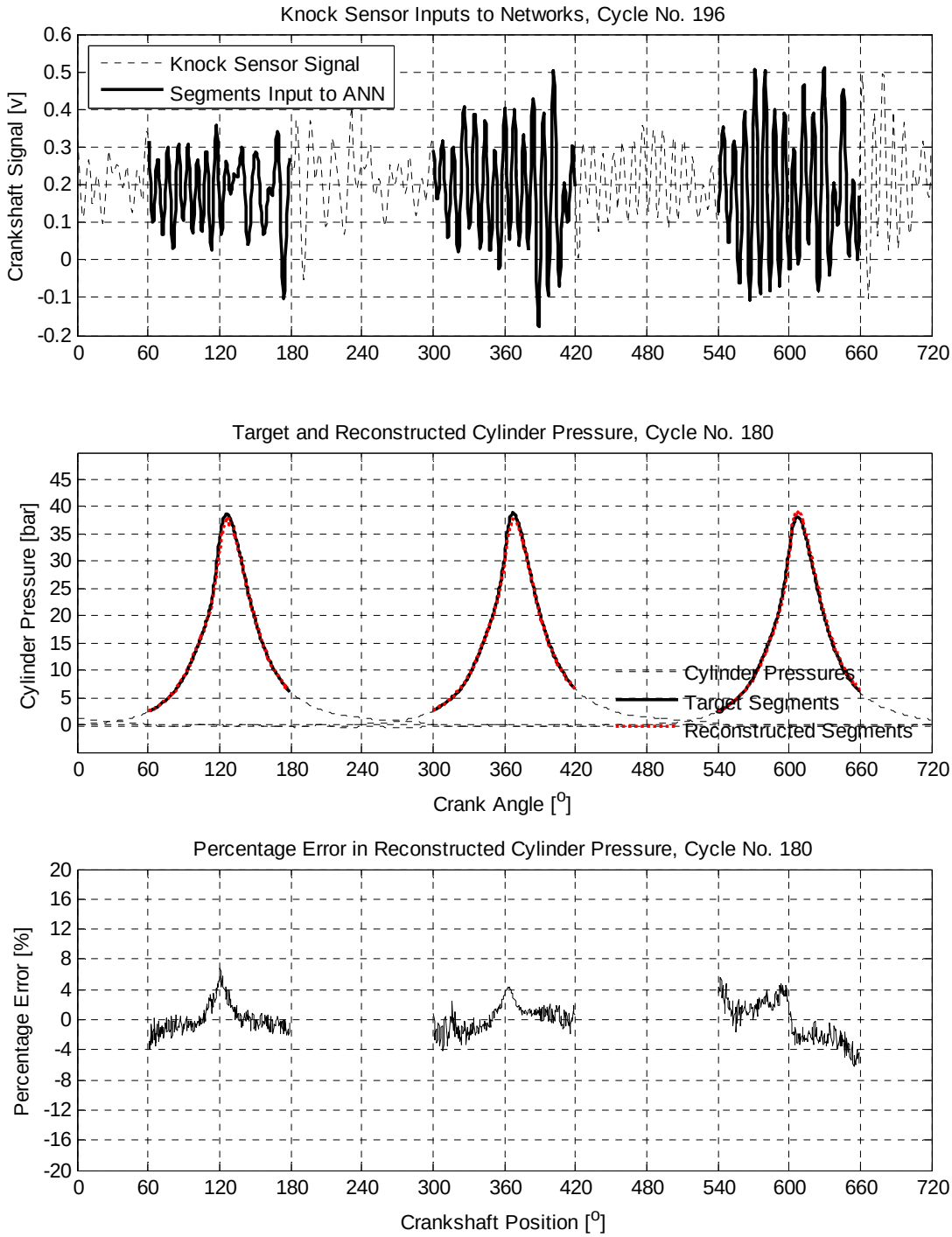


Figure 6.7 – Example cycle of predicted data using unseen crank angle domain knock sensor data as input – 2000 rev/min, 20 Nm
Upper: Knock sensor signal segments used as network inputs
Centre: Target and predicted cylinder pressure segments
Lower: Percentage pressure errors in each reconstructed segment

The reconstructions are for unseen data. The upper data highlights the segments extracted from one cycle for each cylinder as inputs to the network (crank acceleration or knock sensor respectively). The centre graphs

compare the target cylinder pressure segments with the output predictions made by the network. The lower graph shows the percentage error comparing target vs. predicted value – for this example cycle the reconstructed pressures are broadly within 8% of the target using crank acceleration and within 4% of the targets using knock sensor data.

All 198 cycles of data from the unseen dataset were presented to the network and reconstructed cylinder pressures predicted. Figure 6.8 summarises the performance of the network over those cycles with crankshaft acceleration as the input variable. The upper graph shows the percentage error of the predicted magnitude of peak pressure – P_{max}, the centre graph the angular error in prediction of the position of peak pressure relative to TDC firing – θ_{max} and the lower graph the overall Mean Squared Error (MSE) in pressure reconstruction over the window used – 60° BTDC to 60° ATDC. The latter metric gives a measure of the success of reconstruction of the pressure trace shape away from the immediate point of peak pressure. Figure 6.9 shows similar data using crank angle domain knock sensor signals as the inputs for the network.

Table 6.1 summarises the data from Figure 6.8 and Figure 6.9, showing the mean magnitude errors of P_{max} and θ_{max} along with the value for 2 standard deviations of the predicted values – approximately 95% of errors will lie in this range about the mean.

Crank Angle Domain Signal Input to ANN	2000 rev/min, 20 Nm			
	Mean error P_{max} [%]	2σ Error P_{max} [%]	Mean error θ_{max} [^{o}]	2σ error θ_{max} [^{o}]
Knock Sensor	3.2	11.3	1.0	2.0
Crank Acceleration	2.3	4.4	1.2	2.1

Table 6.1 – Summary of errors from crank angle domain training and prediction at 2000 rev/min, 20 Nm

Table 6.1 suggests that crank acceleration results in more reliable estimate of cylinder pressure reconstruction P_{max} than knock sensor inputs for this set of data. θ_{max} estimates are more similar between the two inputs.

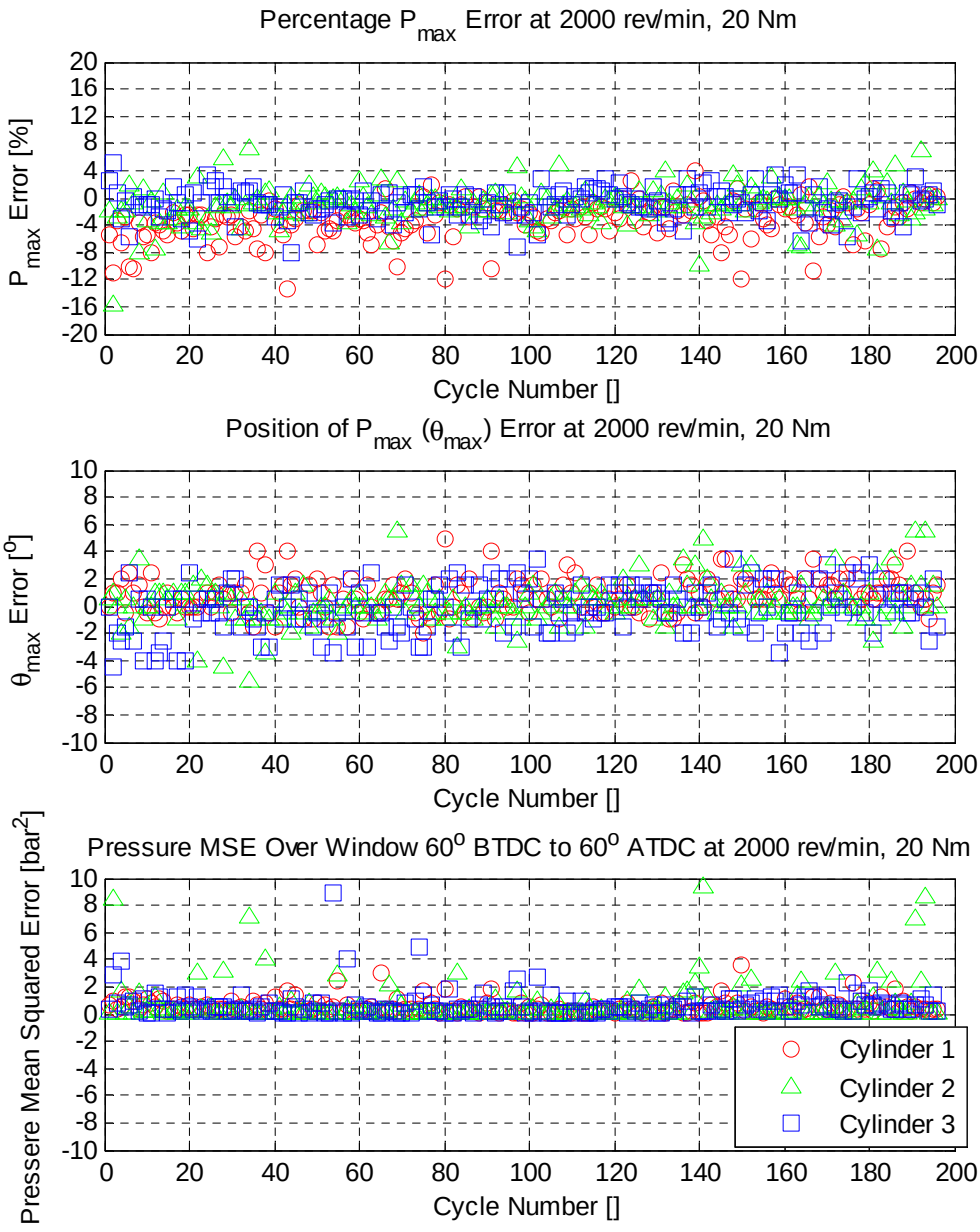


Figure 6.8 – Cylinder pressure reconstruction errors over 195 cycles at 2000 rev/min, 20 Nm (unseen data) using crankshaft acceleration inputs in the crank angle domain
Upper: % P_{max} error
Centre: θ_{max} angle error
Lower: Pressure MSE over data segment 60° BTDC to 60° ATDC

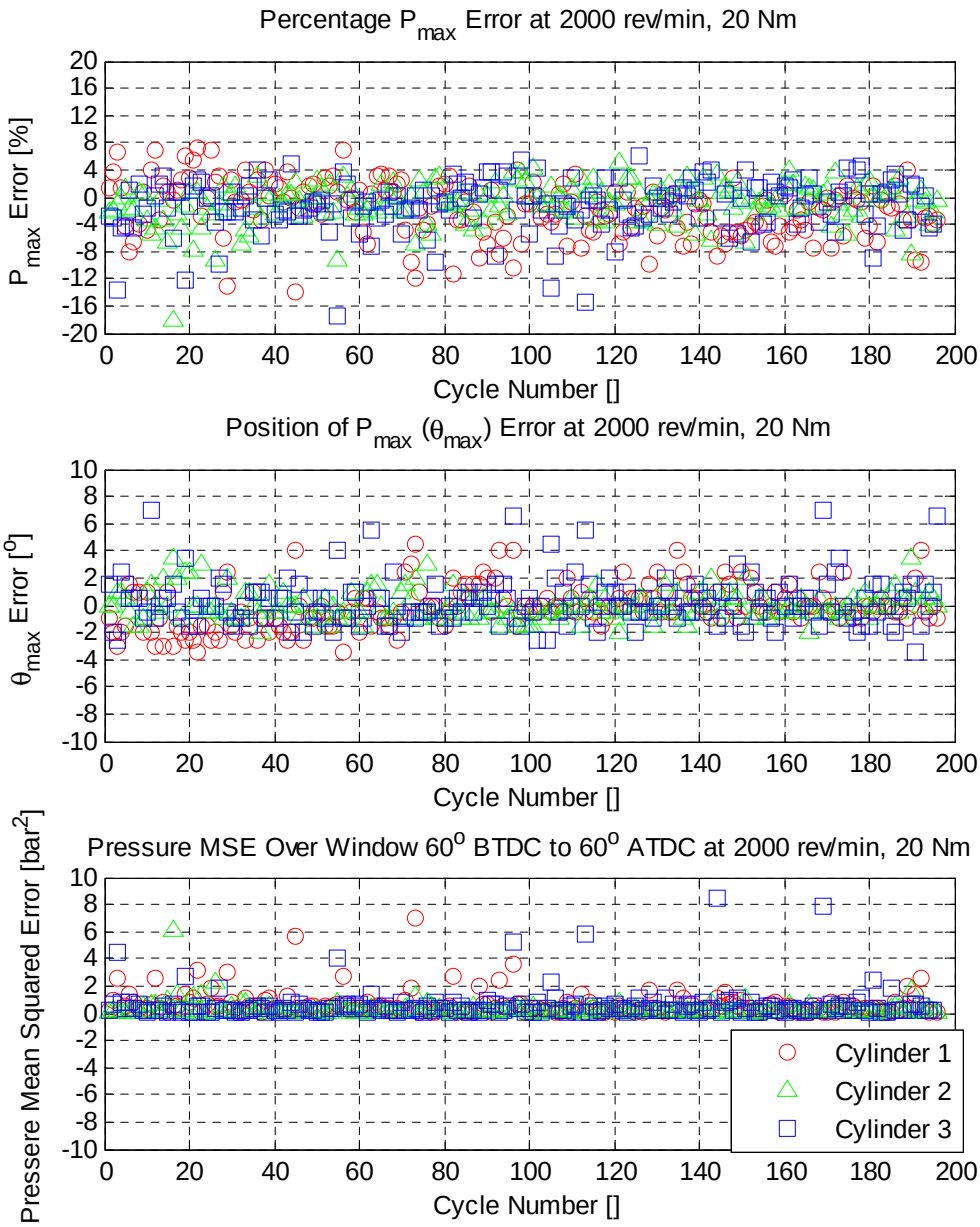


Figure 6.9– Cylinder pressure reconstruction errors over 195 cycles at 2000 rev/min, 20 Nm (unseen data) using knock sensor inputs in the crank angle domain
Upper: % P_{max} error
Centre: θ_{max} angle error
Lower: Pressure MSE over data segment 60° BTDC to 60° ATDC

6.3 Feed Forward Training

A number of engine signals are available in the test datasets that will contain information influenced by the cylinder pressure. To understand what performance may be possible using feedforward architectures only, a series of training exercise were conducted adjusting various network parameters.

The network architecture used was that shown in Figure 6.10, a multi-input, single-output network with a variable number of time delays on the input(s). The inputs could consist of crank acceleration, crank velocity or cylinder block vibration. The output for training targets and network predictions on unseen data was the concatenated cylinder pressure described in section 5.4 – hence the single output architecture.

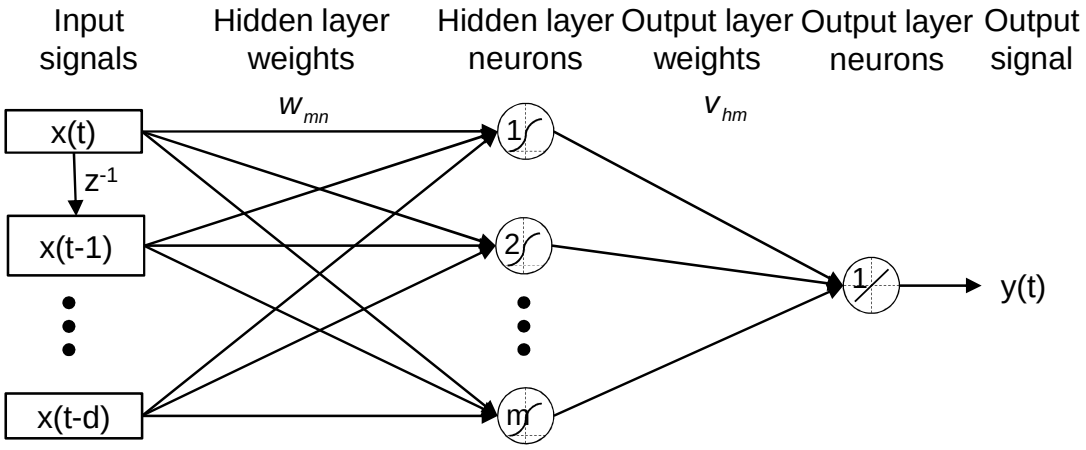


Figure 6.10 – Architecture of feedforward network used for initial benchmark training

The networks were trained using standard MATLAB toolbox functions, based around the time delayed network structures. The parameters varied between different training iterations for the same input / output combinations were:

- Number of engine cycles (and hence 3 x pressure cycles) presented to the network
- Number of delay states on the input vector (i.e. the value of *d* in Figure 6.10)
- Number of neurons in the non-linear hidden layer

Each network was trained 10 times for each set of parameters. Differences between training iterations on the same parameters were seen to be small, and the results presented here are chosen as representative of the typical results for that set of parameters. Each network was trained using data from the 1500 rev/min, 20 Nm operating point dataset.

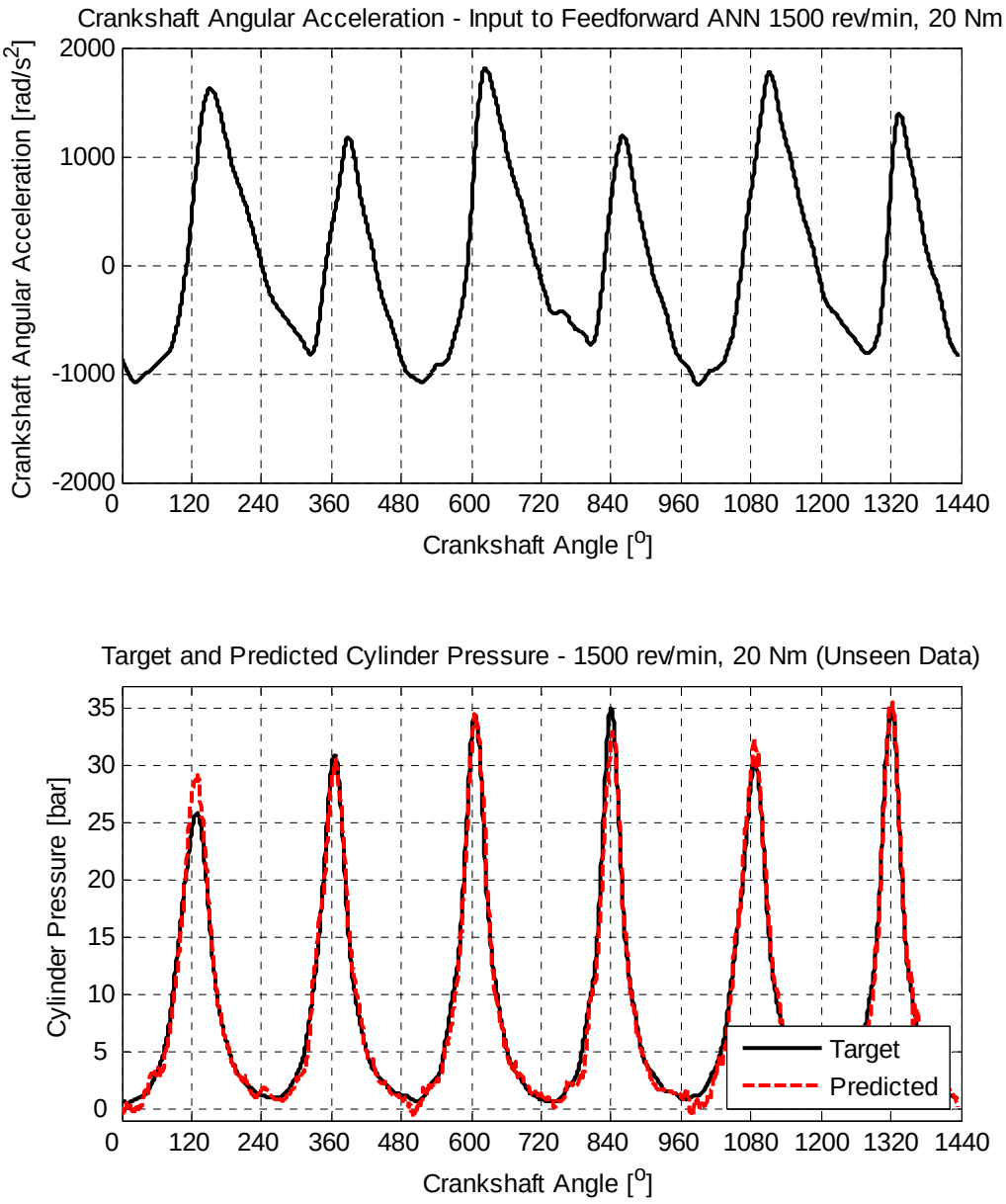


Figure 6.11 – Example of feedforward network at 1500 rev/min, 20 Nm using 20 hidden neurons, 40 input delays and training over 100 cycles
Upper: Crank acceleration input
Lower: Target and predicted cylinder pressures

One dataset was used for training using a varying number of engine cycles from the start of the dataset, according to the first column of Table 6.2. The second data set was then presented as inputs to the network for a prediction on unseen data. The data below all refer to the performance and error during prediction against the unseen dataset.

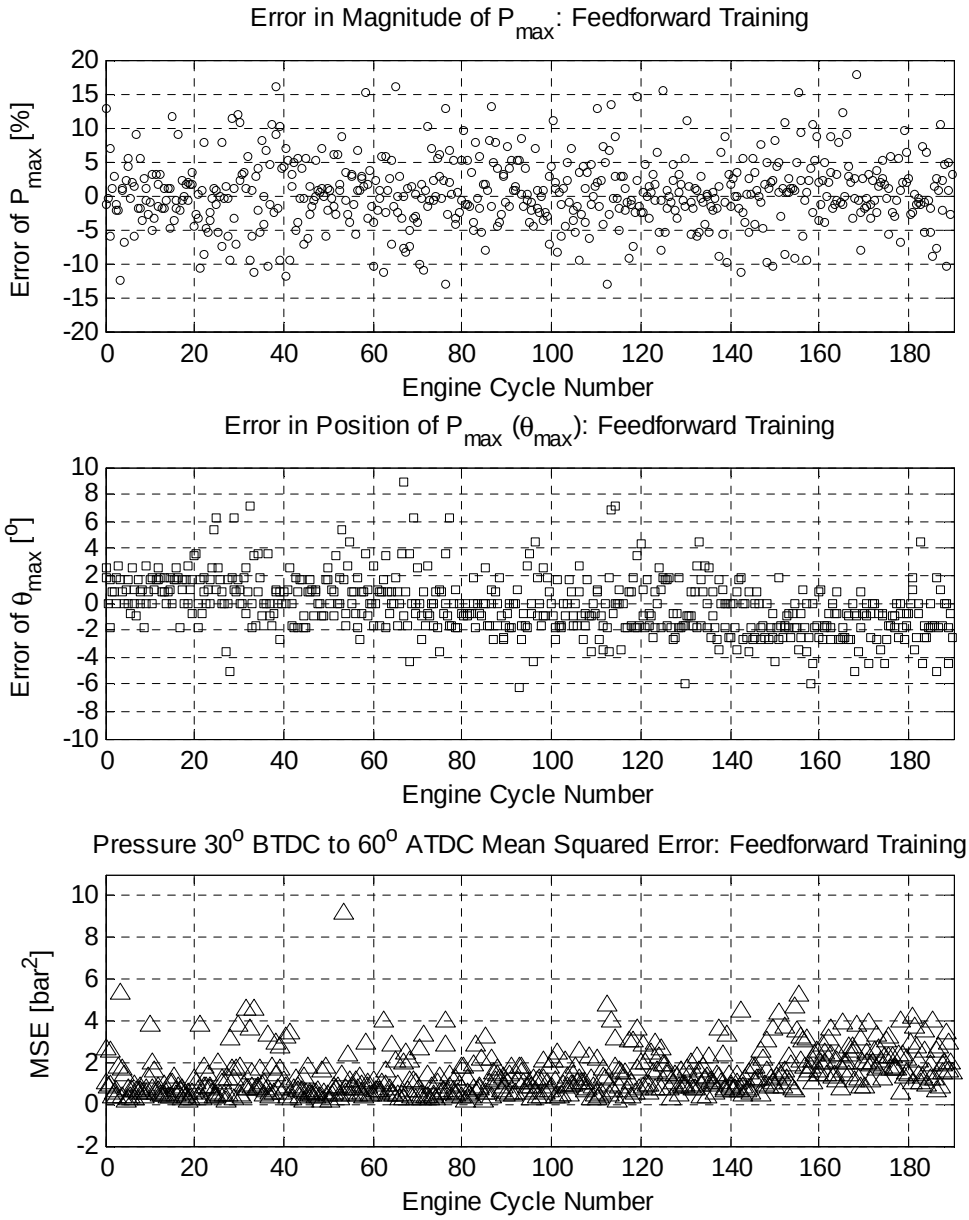


Figure 6.12 – Prediction errors of a feedforward network over 190 engine cycles (3 firing pressure peaks / cycle), 1500 rev/min, 20 Nm
Upper: Error in magnitude of P_{max}
Centre: Error in position of P_{max}
Lower: Pressure MSE over interval 30° BTDC to 60° ATDC

Figure 6.11 shows two typical engine cycles resulting from a feed forward network predicting on unseen data. This network was trained using crankshaft angular acceleration inputs over 100 cycles, and was constructed with 20 hidden neurons and 40 input delays.

Figure 6.12 shows the errors across 190 cycles of prediction, with the metrics for magnitude of P_{max}, θ_{max} and pressure mean squared error (MSE) over the interval 30° BTDC to 60° ATDC for each firing peak pressure event (570 events in total). Table 6.2 summarises the statistics of the errors for each of the sets of network parameters, showing the mean of error magnitudes along with the value of 2 standard deviations – representing the limits around the mean within which approximately 95% of the errors will lie.

Figure 6.11 and Figure 6.12 present data which contribute to row ten of Table 6.2. The performance of this network on unseen data appears to be reasonable. The feed forward architecture offers good stability and the predictions do not deviate to very high or low values. The predicted cylinder pressure follows the pattern of peaks and troughs set by the targets, the mean value of the P_{max} magnitude error is 3.9%, and mean error of the position of P_{max} predictions lie with 1.5° of the target.

	No. of training cycles	No. of hidden Neurons	No of Delay States	P_{max} Error [%]		Θ_{max} Error [^{o}]		MSE Error -30° to +60°	
				Mean	2σ	Mean	2σ	Mean	2σ
1	10	10	10	7.1	12.6	3.5	6.1	5.1	6.3
2	50	10	10	6.7	12.1	3.5	5.4	4.5	5.8
3	100	10	10	5.3	10.3	3.5	5.3	3.9	4.4
4	150	10	10	5.9	11.5	2.7	4.4	3.9	4.9
5	195	10	10	5.7	11.0	2.8	4.6	3.2	5.1
6	100	20	10	6.0	11.2	2.7	5.4	3.4	5.7
7	100	40	10	5.2	10.5	2.5	5.1	2.9	4.8
8	100	60	10	4.8	10.8	2.6	4.4	2.8	4.1
9	100	20	5	6.5	13.2	2.6	4.4	5.1	7.7
10	100	20	20	3.9	7.3	1.5	2.6	1.4	2.8
11	100	20	40	3.9	7.2	1.6	2.4	1.0	1.8

Table 6.2 – Summary performance of varying network parameters for feedforward network - 1500 rev/min, 20 Nm

Inspection of Table 6.2 allows some qualitative conclusions to be drawn about the performance of the network training with variations of the architecture's parameters:

- Increasing the number of engine cycles over which the network is trained initially improves the network, but there is no improvement beyond 100 engine cycles
- Increasing the number of hidden neurons improves the performance, but the improvements are relatively small, and the training times increase significantly
- Increasing the number input delay states improves the performance, but the improvement beyond 20 delays is small

Although not achieving the <4% P_{max} and <1° Θ_{max} errors identified as a successful result, a simple feed forward network does appear capable of a reasonable reconstruction of cylinder pressure from crankshaft acceleration, given some optimisation of network parameters, and when tested on unseen engine data recorded at the same test condition. A more challenging test is to ask the network to take unseen inputs from a different speed and load operating condition.

The initial training and reconstruction was conducted at the mid-point of the dataset matrix – 1500 rev/min and 20 Nm. Crank acceleration data from two other test conditions was then presented to the network and pressure reconstructions calculated over 190 cycles. Figure 6.13 shows two cycles of the reconstructed trace against the target at 1000 rev/min, 10 Nm and Figure 6.14 shows similar data but for pressures reconstructed at 2000 rev/min, 30 Nm. Table 6.3 summarises the error statistics for each firing pressure peak across the 190 cycles.

It is immediately clear that the network is less successful at reconstructing cylinder pressure at different speed and load conditions from that at which it was trained. However, a reasonable shape of pressure curve is still predicted, with θ_{max} broadly in the correct part of the cycle. The network is always stable, but this is expected given there is no feedback in the system.

The lower speed, lower load condition generally under predicts P_{max}, while the higher speed, higher load condition shows a particular loss of prediction accuracy around P_{max} on cylinder number 1, the predictions for cylinders 2 and 3 being significantly better. There is some obvious loss of accuracy away from P_{max} locations too, though these regions of the cycle are of lower importance.

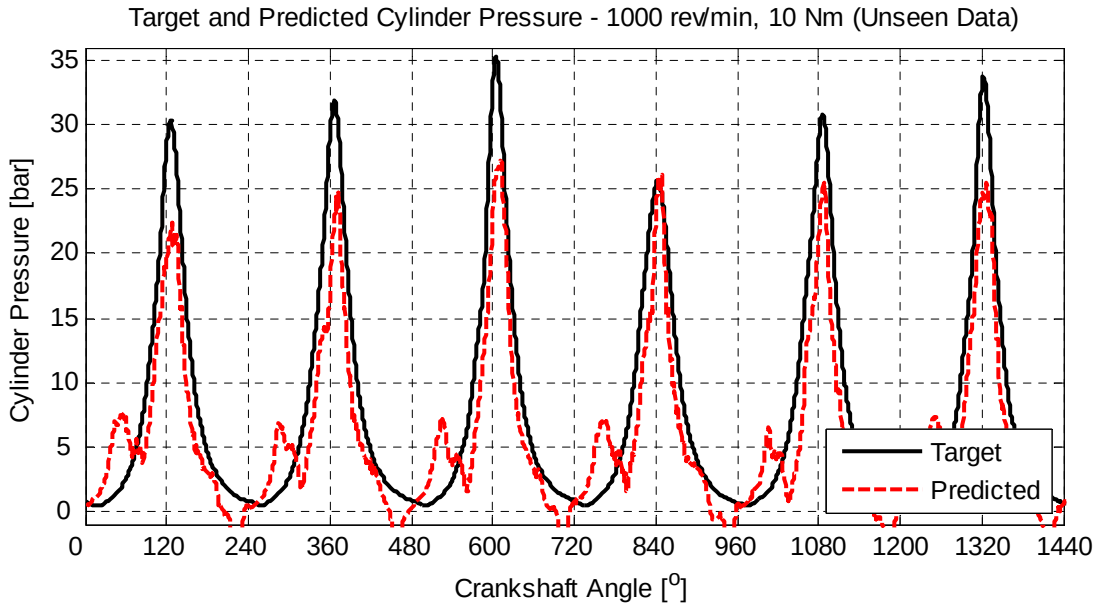


Figure 6.13 – Target and reconstructed pressures at 1000 rev/min, 10 Nm, using a feedforward network trained at 1500 rev/min, 20 Nm

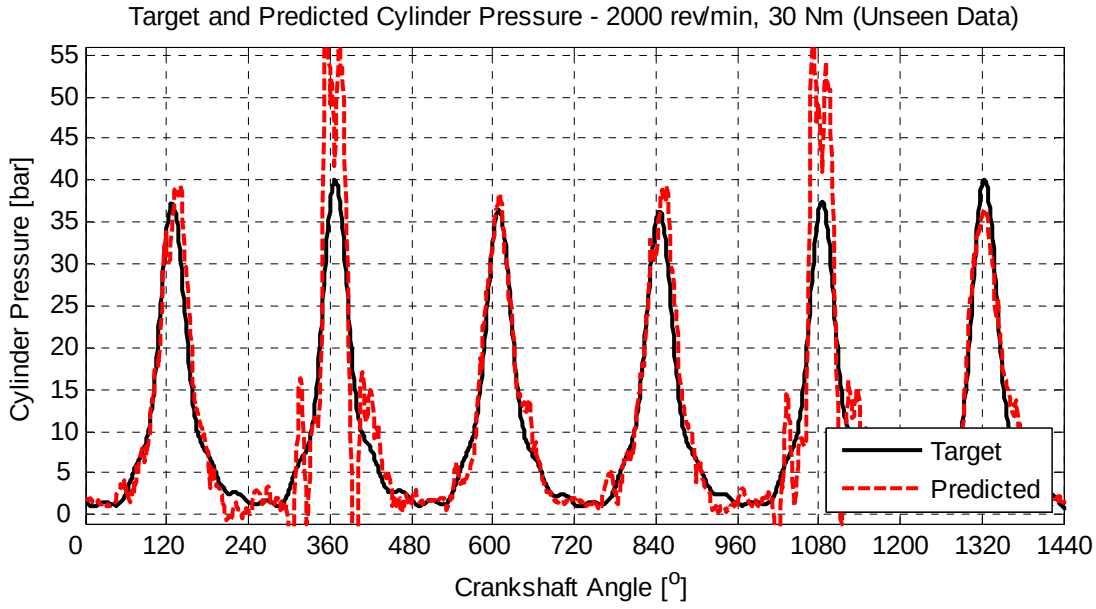


Figure 6.14 – Target and reconstructed pressures at 2000 rev/min, 30 Nm, using a feedforward network trained at 1500 rev/min, 20 Nm

Comparing the error statistics values from Table 6.3 with those taken from row 10 of Table 6.2 confirms that performance measured by the metrics around P_{max} are significantly worse for reconstruction at operating conditions away from the training condition.

Generalisation Condition	P_{max} Error [%]		Θ_{max} Error [^{o}]		MSE Error -30° to +60°	
	Mean	2σ	Mean	2σ	Mean	2σ
1500 rev/min, 20 Nm (training condition)	3.9	7.3	1.5	2.6	1.4	2.8
1000 rev/min, 10 Nm	20.3	12.6	3.3	5.6	30.1	15.4
2000 rev/min, 30 Nm	18.7	41.3	7.5	11.9	38.5	90.6

Table 6.3 – Error statistics for feedforward network predicting pressure at conditions different to the training condition

6.4 Teacher Forced Recurrent Training of NARX Architectures

Figure 6.15 shows the layout of a multi input, single output RNN potentially suitable for use in cylinder pressure reconstruction. The input vector would comprise of one or more engine signals *x(t)*, optionally with some time delay states *x(t-1)* to *x(t-d)* and a recurrent input fed back from the previous time step prediction of the output, with some number of delay states *y(t-1)* to *y(t-l).* The challenge is to have a stable training method when the next input vector is recursively dependant on the output.

One approach, straight forward to implement, is the use of teacher forcing during training. As the target values are already known for the training dataset, it is feasible to construct a feed forward network using previous target time step values in place of *y(t-1)* to *y(t-l).* The network may then be trained using the standard back-propagation algorithm in just the same manner as could be used for any other feed forward architecture. Figure 6.16 shows the recurrent network from Figure 6.15 re-configured to a feed forward structure suitable for teacher forced training with known target values (appropriately time-delayed) in the input vector.

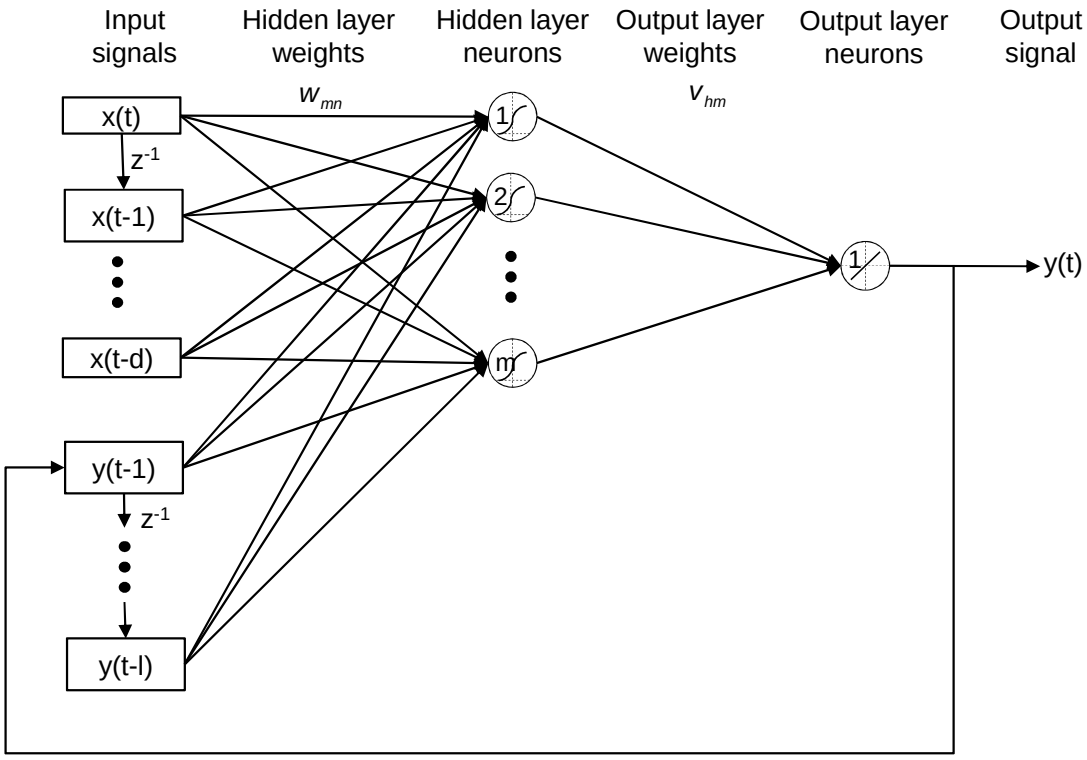


Figure 6.15 – Recurrent Neural Network in NARX structure

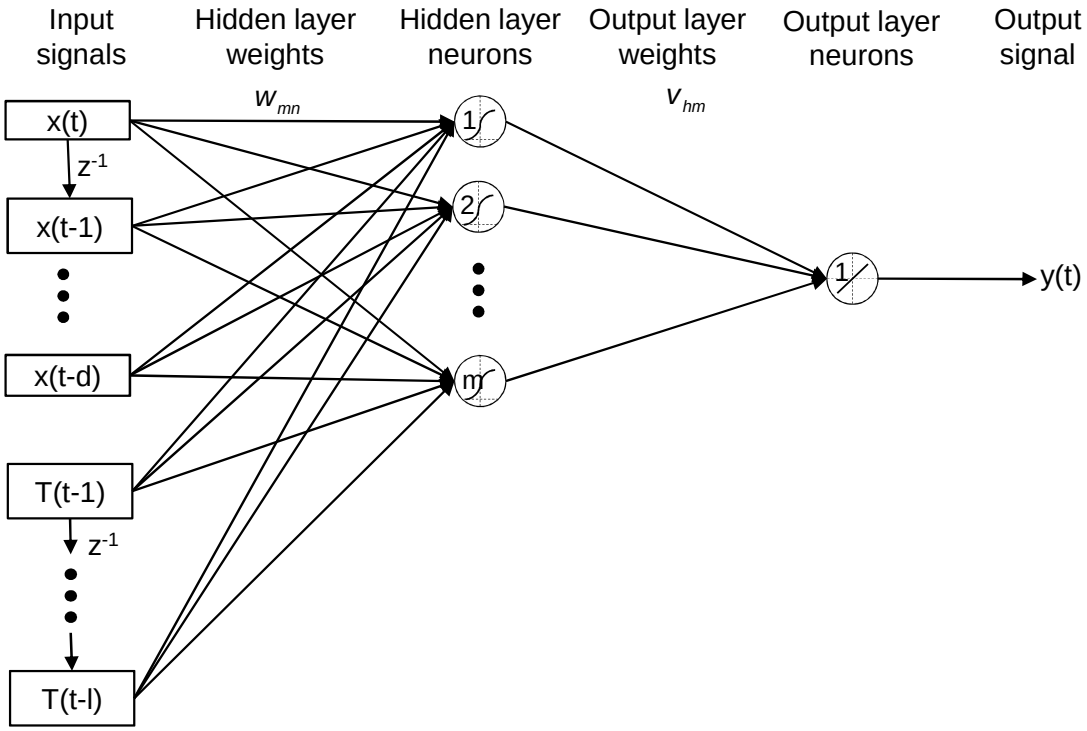


Figure 6.16 – Recurrent structure re-configured as teacher forced feed forward network for training with known time delayed target values replacing recurrent line in input vector (T = training target values)

Experiments with teacher forcing using crankshaft acceleration as the exogenous input showed mixed results. After training, keeping the network in the teacher forced configuration and making predictions with the feedforward network using the static weights resulting from training invariably produced very good results, both on the training data and on unseen data from the same operating condition. Indeed, this configuration usually also gave good prediction when input data came from a different operating condition than that used for training. This is promising, as it suggests that if predictions from the network are good, then next step prediction could continue be good too.

However, there is a pitfall with the type of data that is being input to the network for this particular problem. The frequency content of the cylinder pressure signal is significantly lower than the rate at which the data is sampled (as discussed in section 5.2). This must mean that the target for the next sample prediction from the network at time *t* can never be very different from the (*t-1*) teacher forced input. Hence it is likely (depending on just where the randomised initial weights cause the network to begin in weight space) that the weights will develop during training to propagate the teacher forced inputs as important factors in the network prediction. If this is the case, then once any significant inaccuracy of the prediction occurs, the network will follow this, and is likely to lose control of a stable prediction. Attempts to run the teacher forced networks in NARX configuration, were not successful – frequently the prediction would diverge immediately to a high or low value, occasionally a part of the first cycle would be predicted reasonably well before divergence occurred.

Mass teacher forced network training

The variable behaviour of NARX prediction on teacher forcing trained networks lead to a background effort to run a large number of teacher forced training examples to see whether there was ever likely to be a good result – the anticipated conclusion being that it was not a method to pursue further. A MATLAB script was prepared to repeatedly train and test over the operating condition matrix, producing a brief performance report on NARX prediction at the end of each one.

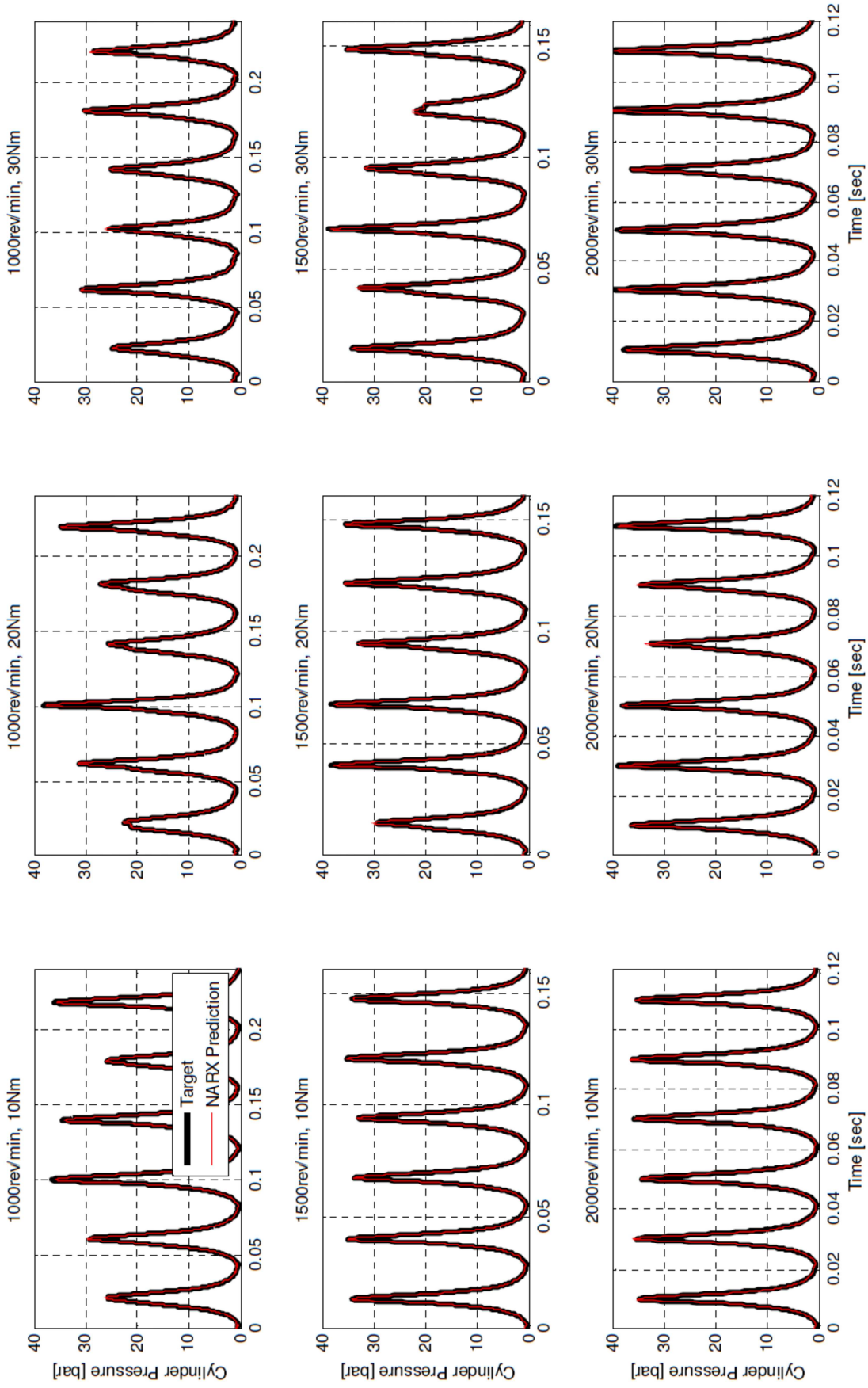


Figure 6.17 – NARX predictions resulting from a single good example network from a mass repeat training exercise

After more than one thousand iterations of training, one example was found to have resulted in good NARX prediction – generalising well at all operating conditions after training on the 1500 rev/min, 30 Nm condition. P_{max} errors lay within 2.5% and θ_{max} within 2.5° of the targets. No network information was stored during this mass exercise, and the results are not repeatable. This is in no way suggested as a potential approach to training NARX networks at it offers no reliability, but it does offer some evidence that the NARX architecture can be capable if robust and reliable training can be achieved. Figure 6.17 shows example cycles from the exercise.

Teacher forced training with noise corrupted input

In an attempt to reduce the potential dependency of a teacher forced network on the accuracy of the forced inputs, a modification to those forced inputs was made by adding noise to the forced signal (and hence also to any time delayed inputs of that signal). The target values were not corrupted by noise.

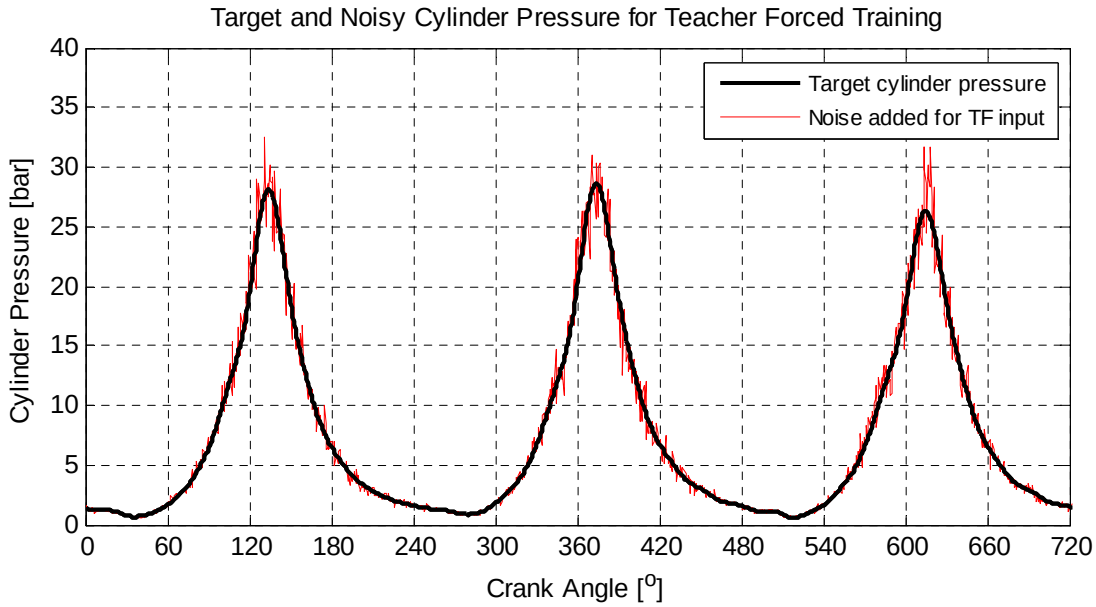


Figure 6.18 – Target cylinder pressure, and noise corrupted signal for teacher forced input

Figure 6.18 shows the target and noise corrupted version of the signal used for the teacher forced input. The noise corruption is created by multiplying the target signal by an equal length synthesised signal consisting of normally distributed random values with mean 1.0 and standard deviation 0.1.

Effectively the target signal is randomly corrupted with a standard deviation of 10%.

Training using this input was based on the direction from section 6.3 using 20 delays on both the exogenous crankshaft acceleration signal and the teacher forced, corrupted target data. The network had 20 hidden neurons, and was trained over 100 engine cycles using data from the 1500 rev/min, 20 Nm operating condition.

After training, predictions were made on the training data set:

i) Teacher forced, feedforward, with the forced inputs the same corrupted signals as used for training

ii) NARX configuration

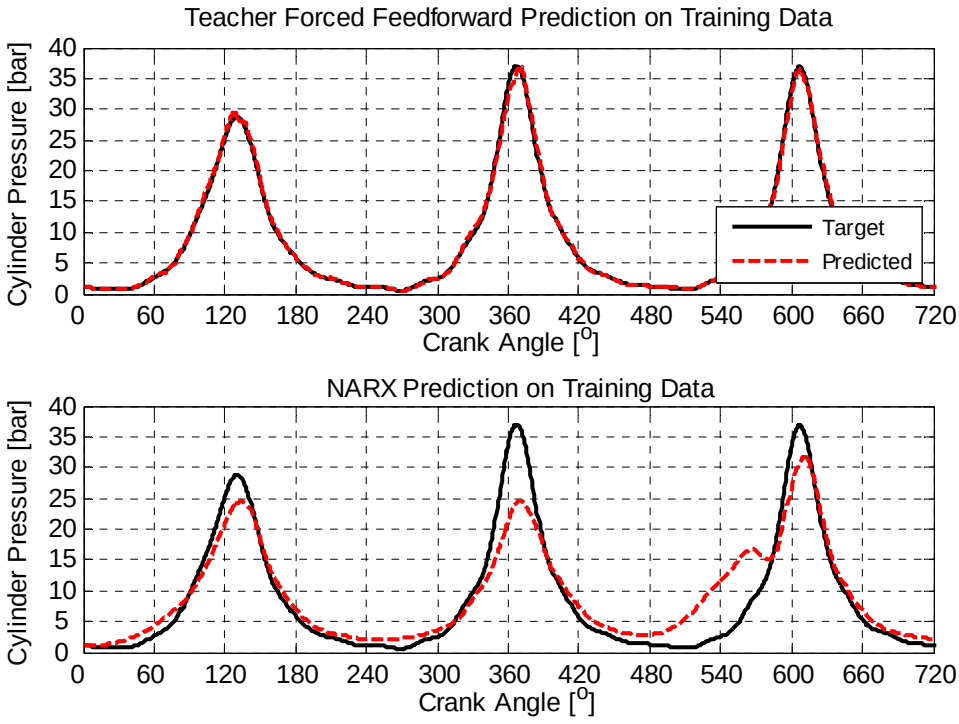


Figure 6.19 – Teacher forced and NARX predictions on training data at 1500 rev/min, 20 Nm

Results for a typical engine cycle for these predictions are shown in Figure 6.19. Despite the forced input being corrupted with noise, the teacher forced, feedforward prediction achieved very good reconstruction of cylinder pressure. This result is promising, as this suggests network may now be

more resilient to error on previous time step predictions. The true NARX prediction is not as accurate, but is better than most networks trained without noise corruption, and in particular does not become unstable anywhere over the 100 engine cycles of the full prediction.

Similar predictions were conducted for the unseen dataset at the same operating condition, once more corrupting the target cylinder pressure before its application to the forced inputs of the network. Figure 6.20 shows the results.

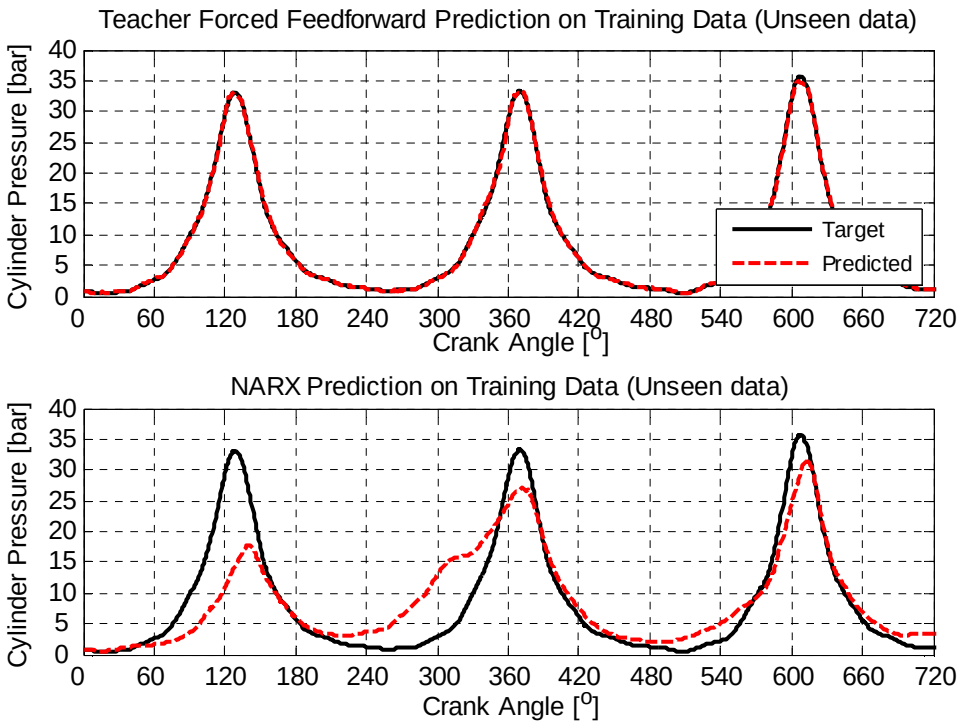


Figure 6.20 – Teacher forced and NARX predictions on unseen data at 1500 rev/min, 20 Nm

Once again, the teacher forced prediction with noisy forced inputs is good. The NARX prediction is less good than the training data example, but crucially still remains stable, and has a broadly credible shape.

Having trained and tested with data from the 'mid-point' of the operating matrix (1500 rev/min, 20 Nm), the network was then tested to a similar procedure using data from operating conditions at the 'corners' of the matrix. For each test, the forced inputs were noise corrupted cylinder pressures

signals, with the same noise statistics as used for the training exercise. Results with a typical cycle for each test are shown according to:

- 1000 rev/min, 10 Nm: Figure 6.21
- 1000 rev/min, 30 Nm: Figure 6.22
- 2000 rev/min, 10 Nm: Figure 6.23
- 1000 rev/min, 30 Nm: Figure 6.24

In each case, the forced, feedforward prediction is quite good. The curve shape is much as it should be, and the magnitude and location of peak pressure is qualitatively close to the target.

The quality of the NARX predictions are much more variable. The two 10 Nm operating cases lose stability almost immediately and do not recover, while the two 30 Nm operating cases remain stable, but the curve shapes are somewhat variable – 2000 rev/min, 30 Nm is the best of the NARX predictions, and here the shapes are reasonably representative.

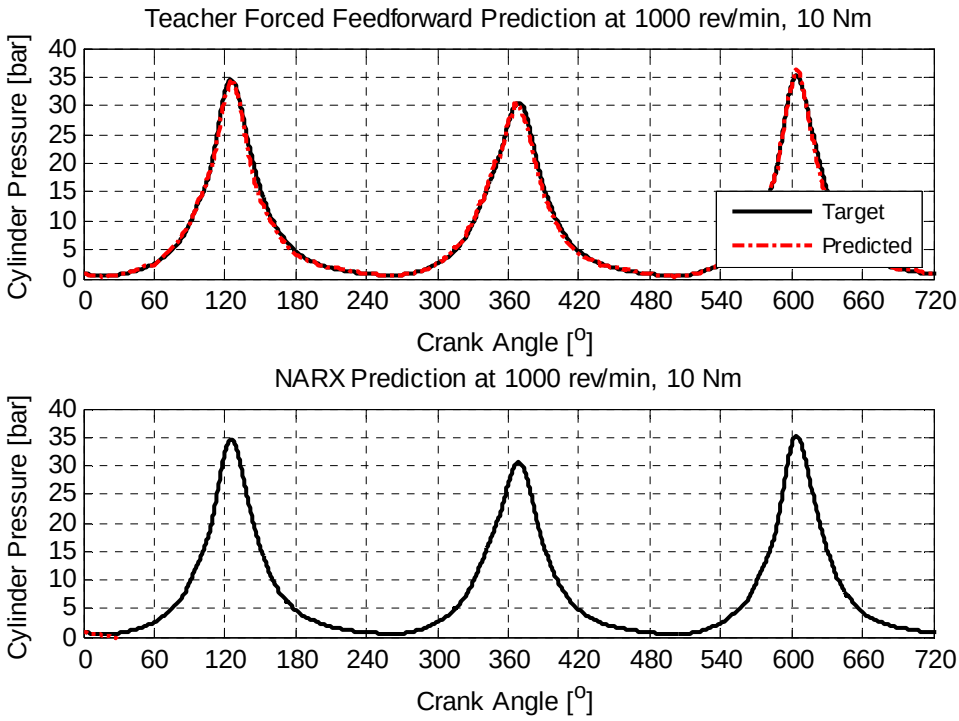


Figure 6.21 – Forced and NARX predictions at 1000 rev/min, 10 Nm

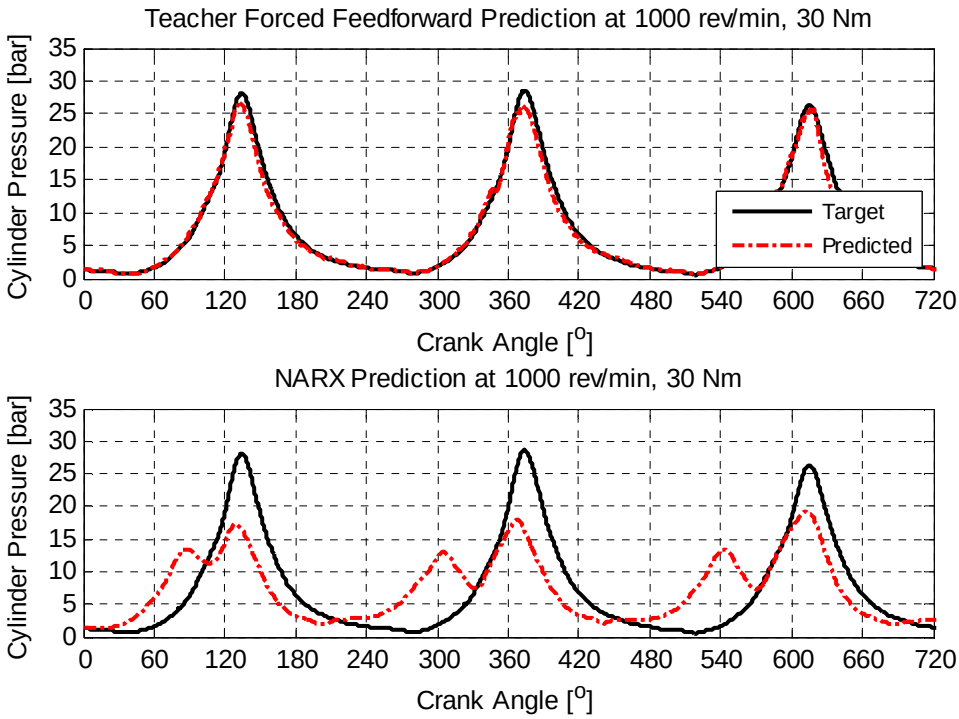


Figure 6.22 – Forced and NARX predictions at 1000 rev/min, 30 Nm

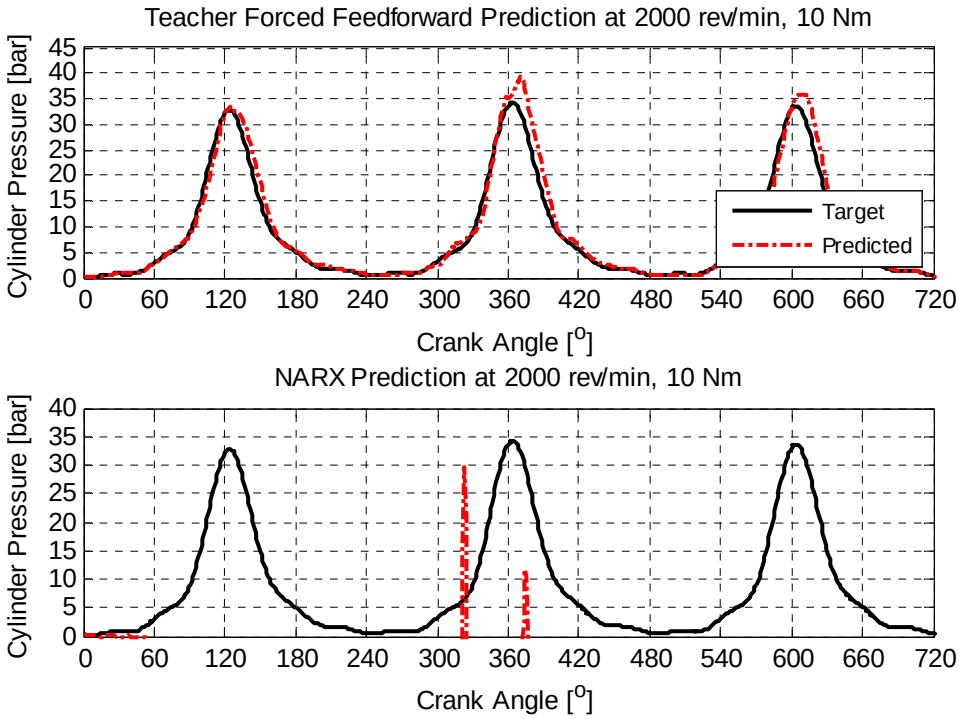


Figure 6.23 – Forced and NARX predictions at 2000 rev/min, 10 Nm

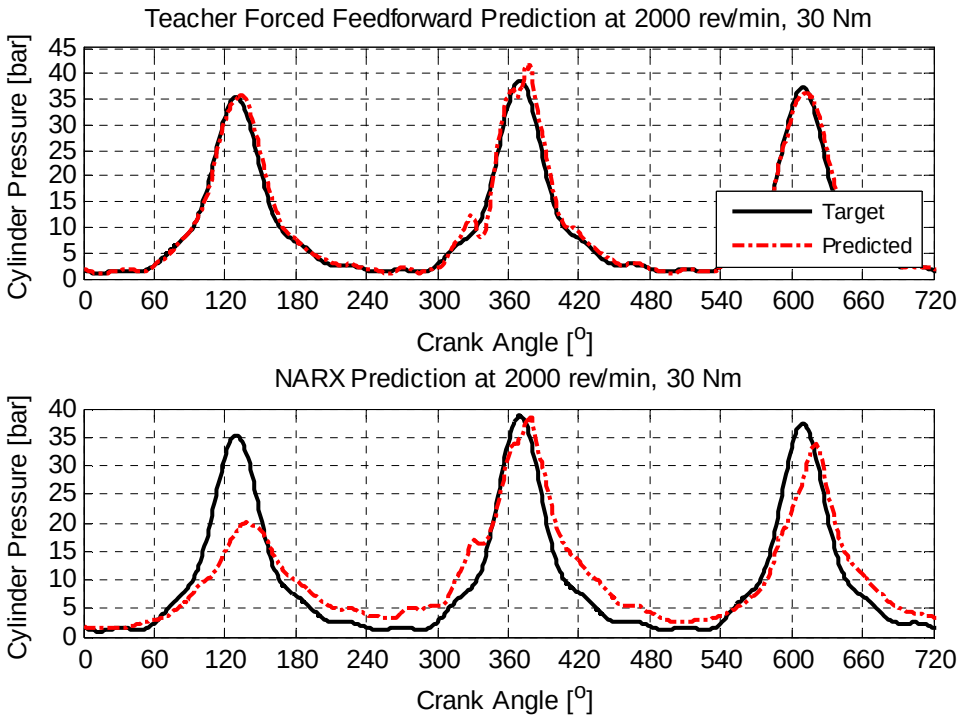


Figure 6.24 – Forced and NARX predictions at 2000 rev/min, 30 Nm

6.5 Conclusions of Feedforward Training Investigations

Feedforward training of ANNs has offered some interesting conclusions.

i) Although the main thrust of this thesis is based around time domain reconstruction, results of angle domain reconstruction have been seen to be quite reasonable on both crank kinematics and structural vibration inputs, and in line with other researchers' findings.

ii) Feedforward training and reconstruction on crank kinematic data has been investigated and gives results that have mean errors that are close to requirements when tested at the same operating condition as used for training. Predicting with the networks at operating conditions different to those used for training produces much less useful predictions, but remains naturally stable.

iii) Teacher forced training of a NARX architecture shows some promise, and offers encouragement that the architecture is worthy of further effort.

iv) One very successful network giving good NARX predictions resulted from a mass teacher forced training exercise.

v) A teacher forced network using a noise corrupted version of the target signal as the forced input trained successfully. This network predicted well in a feed forward, forced architecture using more noise corrupted input data from all operating conditions. This does suggest the NARX architecture is capable of rejecting prediction errors in the recurrent feedbacks if correctly trained.

vi) NARX predictions of the networks trained with noise corrupted teacher forcing data were not very successful in error terms, but were at least stable when predicting at the same operating condition as training, and also stable at some of the other operating conditions – this stability being significantly better than the majority of networks teacher forced trained without noise corruption.

Chapter Seven

7. IMPLEMENTATION AND TESTING OF THE ROBUST ADAPTIVE GRADIENT DESCENT ALGORITHM – A FULLY RECURRENT TRAINING STRATEGY FOR NARX NETWORK CYLINDER PRESSURE RECONSTRUCTION

The conclusion reached by previous researchers [Potenza et al. 2007] and supported by the work in chapter 6 is that the NARX architecture for a Recurrent Neural Network (RNN) offers promise for cylinder pressure reconstruction, but that a reliable, robust and efficient method of training is required.

Previous work within the research project found training via Back Propagation Through Time (BPTT) and the Extended Kalman Filter (EKF) [Potenza 2005], [Vulli 2006] to offer promise, but to be inefficient. A literature survey of recent developments in training RNNs, resulted in the choice of the Robust Adaptive Gradient Descent (RAGD) algorithm as the most promising for the time series data and NARX architectures under consideration. Particularly, the network structure used for the method's development is directly applicable to the NARX structures described in section 6.4 using the combined single pressure from section 5.4 to make the network single output. The promised stability of the algorithm is also attractive. Ultimately the project chose to pursue two recurrent training methods, RAGD and the Optimal Bounded Ellipsoid method described by [Rubio et al. 2009].

The algorithm developed in a paper by [Song et al. 2008] describes the method in detail, focusing on a multi-input, single output RNN with a variable number of recurrent feedback lines that is well suited to the signal structures identified through this thesis, in particular using the concatenated cylinder pressure traces described in section 5.4 as the target values. This allows multi-cylinder engine pressure traces to be combined into a single target output, and hence significantly simplifying the network, the implementation of the training algorithm, and speeding the training execution. The authors detail the guaranteed stability of the method, and other papers by the authors

[Wu et al. 2008] extend the approach to multi input – multi output implementations and to further applications of prediction and control.

A description of the method is reproduced from [Song et al. 2008] here, as implemented for cylinder pressure reconstruction in MATLAB. Sections 7.1, 7.2 and 7.3 reproduce the method's equations directly from [Song et al. 2008] with only minor changes of symbols to match the resulting MATLAB implementation. Aspects of the implementation are then discussed, and the results achieved with the method operating on engine test data are described.

7.1 Robust Adaptive Gradient Descent Algorithm (RAGD)

The RAGD training algorithm implemented for cylinder pressure reconstruction is based on the description in [Song et al. 2008]. Figure 7.1 shows the network structure under consideration.

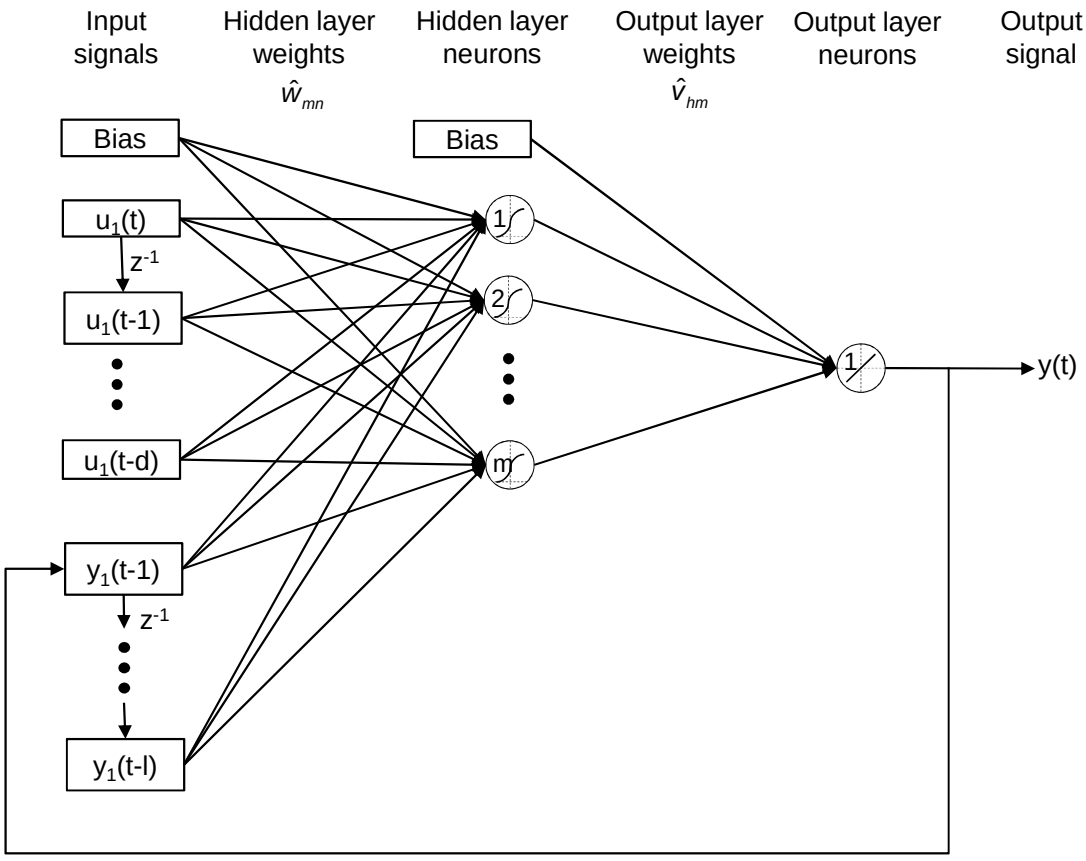


Figure 7.1 – Diagram of multi-input, single-output (MISO) RNN for RAGD training method

The input vector *x* in general has *n* elements, consisting of a bias, exogenous inputs *u* each with a value at the current step and some number of time delays *d.* For cylinder pressure reconstruction, these exogenous inputs will consist of crank acceleration, knock sensor signals etc. The input vector also includes *l* time delays of feedback from previous network output predictions.

A single constant input value of unity acts as the bias for all hidden neurons, with the bias trained as an additional column of the hidden weight matrix. The input vector has the form $x \in \mathbb{R}^{nx1}$ where $n = 2 + d + l$.

The hidden layer consists of *m* neurons, with sigmoid activation functions denoted as $h(\bullet)$. The function being described by as follows:

$$h(\bullet) = \frac{1}{1+e^{-\lambda \bullet}} \tag{7.1}$$

The value λ is implemented as a variable, but remains at unity for all training examples discussed in this thesis.

The hidden layer weight $\hat{W}$ matrix has the form $\hat{W} \in \mathbb{R}^{mxn}$. The output layer of this MISO network has a single linear neuron, and the output layer weight matrix $\hat{V}$ is of the form $\hat{V} \in \mathbb{R}^{1xn}$. Once more, a single constant input value of unity acts as the bias for the output layer, trained as an additional column of the output weight matrix.

The single neuron output from the network is $y(t)$, evaluated at any time step *t* by equation (7.2) which is the standard forward propagation evaluation:

$$y(t) = \hat{V}(t)\,\phi\left(\hat{W}(t)x(t)\right) \tag{7.2}$$

The vector of values output by the hidden layer, is the second product term of the right side of equation (7.2), evaluated as follows (and written as $\phi(t)$ in the remainder of this chapter):

$$\begin{aligned} \phi(t) &= \phi\left(\hat{W}(t)x(t)\right) \\ &= \left[h\left(\hat{W}_{1,:}(t)x(t)\right)\ h\left(\hat{W}_{2,:}(t)x(t)\right)\cdots h\left(\hat{W}_{m,:}(t)x(t)\right)\right]^T \end{aligned} \tag{7.3}$$

Where the superscript *T* represents the transpose of the vector.

During training, the target output values (desired values) are denoted by $d(t)$, and the prediction error $e(t)$, is:

$$e(t) = d(t) - y(t) + \varepsilon(t) \tag{7.4}$$

where $\varepsilon(t)$ represents a disturbance value

The cost function for the RAGD algorithm is taken as:

$$E(t) = \frac{e^2(t)}{2} \tag{7.5}$$

This is convenient and fast for online learning as the derivative of $E(t)$ with respect to time step evaluates simply to $e(t)$. The training objective is to update the hidden and output layer weight matrices such that $E(t)$ is minimised.

7.2 Output Layer Weight Matrix Update

The RAGD algorithm uses an adaptive hybrid learning algorithm, working in both standard online Back-Propagation (BP) and Real Time Recurrent Learning (RTRL) manners according to stability and convergence conditions. The output weight update equation is given as:

$$\begin{aligned} \hat{V}(t+1) &= \hat{V}(t) - \frac{\alpha^v(t)}{\rho^v(t)} \frac{dE(t)}{d\hat{V}(t)} \\ &= \hat{V}(t) - \frac{\alpha^v(t)}{\rho^v(t)} e(t) \frac{dy(t)}{d\hat{V}(t)} \end{aligned} \tag{7.6}$$

The estimated derivative of the network's output with respect to the output layer weights is defined as follows:

$$\begin{aligned} \frac{dy(t)}{d\hat{V}(t)} &= \frac{\partial y(t)}{\partial \hat{V}(t)} + \beta^v(t) \frac{\partial y(t)}{\partial x(t)} \frac{\partial x(t)}{\partial \hat{V}(t)} \\ &= \phi(t)^T + \beta^v(t) \hat{A}(t) \end{aligned} \tag{7.7}$$

Substituting equation (7.7) in to equation (7.6) , we have:

$$\hat{V}(t+1)=\hat{V}(t)+\frac{\alpha^{v}(t)}{\rho^{v}(t)}\cdot e(t)\cdot\left[\phi(t)^{T}+\beta^{v}(t)\cdot\hat{A}(t)\right] \tag{7.8}$$

where $\alpha^{v}(t)$ is an adaptive learning rate defined as:

$$\begin{aligned}
&\alpha^{v}(t)=1 \qquad \text{if} \quad |e(t)|\geq\varepsilon_{m}^{v}\Big/\sqrt{1-\frac{\left\|\phi(t)^{T}+\beta^{v}(t)\hat{A}(t)\right\|^{2}}{\rho^{v}(t)}}\\
&\alpha^{v}(t)=0 \qquad \text{if} \quad |e(t)|<\varepsilon_{m}^{v}\Big/\sqrt{1-\frac{\left\|\phi(t)^{T}+\beta^{v}(t)\hat{A}(t)\right\|^{2}}{\rho^{v}(t)}}
\end{aligned} \tag{7.9}$$

and where $\varepsilon_{m}^{v}=\max\left(\left|\tilde{\varepsilon}^{v}(t)\right|\right)$, and $\rho^{v}(t)$ is a normalisation factor such that:

$$\rho^{v}(t)=v\,\rho^{v}(t-1)+\max\left\{\bar{\rho}^{v},\left\|\phi(t)^{T}+\beta^{v}(t)\hat{A}(t)\right\|^{2}\right\} \tag{7.10}$$

with $0<v<1$ and $0<\bar{\rho}^{v}$ are positive constants.

The variable $\beta^{v}(t)$ in equations (7.9) and (7.10) is a hybrid adaptive learning rate defined by:

$$\begin{aligned}
&\beta^{v}(t)=1, \qquad \text{if} \quad \phi^{T}(t)\left\{\delta I+\phi(t)\phi^{T}(t)\right\}^{-1}\hat{A}^{T}(t)\geq 0\\
&\beta^{v}(t)=0, \qquad \text{if} \quad \phi^{T}(t)\left\{\delta I+\phi(t)\phi^{T}(t)\right\}^{-1}\hat{A}^{T}(t)<0
\end{aligned} \tag{7.11}$$

with δI a small positive constant.

Furthermore $\hat{A}(t)$ is the extended recurrent gradient of the form $\hat{A}(t)\in\mathbb{R}^{1xm}$:

$$\hat{A}(t)=\hat{V}(t)\phi'(t)\hat{W}(t)\hat{D}_{v}(t) \tag{7.12}$$

here $\hat{D}_{v}(t)$ is the Jacobian matrix such that $\hat{D}_{v}(t)\in\mathbb{R}^{nxm}$, defined as:

$$\begin{aligned}
\hat{D}_{v}(t)&=\left[\begin{array}{cccccc}\frac{\partial u(t)}{\partial\hat{V}(t)} & \cdots & \frac{\partial u(t-d)}{\partial\hat{V}(t)} & \frac{\partial y(t-1)}{\partial\hat{V}(t)} & \cdots & \frac{\partial y(t-l)}{\partial\hat{V}(t)}\end{array}\right]\\
&\approx\left[\begin{array}{cccccc}\frac{du(t)}{d\hat{V}(t)} & \cdots & \frac{du(t-d)}{d\hat{V}(t)} & \frac{dy(t-1)}{d\hat{V}(t-1)} & \cdots & \frac{dy(t-l)}{d\hat{V}(t-l)}\end{array}\right]
\end{aligned} \tag{7.13}$$

Normally the exogenous inputs and the bias do not change with the weight matrix, so the first $d+2$ entries of $\hat{D}_{v}(t)$ are zero vectors.

Finally, $\phi'(t)$ is a diagonal matrix of the form $\phi'(t) \in \mathbb{R}^{mxm}$ constructed as:

$$\phi'(t) = diag\left[\ \phi_1'(t)\ \ \phi_2'(t)\ \ \cdots\ \ \phi_m'(t)\ \right] \tag{7.14}$$

Equations (7.9) to (7.14) provide all elements needed to evaluate equation (7.8) such that the output layer weight matrix may be updated.

7.3 Hidden Layer Weight Matrix Update

The hidden layer weight matrix update uses an adaptive normalised gradient algorithm constructed in a similar manner to that for the output layer, though some aspects are more complex as there are many hidden neurons compared to a single output neuron. The hidden layer weight update is given as:

$$\begin{aligned}\hat{W}(t+1) &= \hat{W}(t) - \frac{\alpha^w(t)}{\rho^w(t)}\frac{dE(t)}{d\hat{W}(t)} \\ &= \hat{W}(t) - \frac{\alpha^w(t)}{\rho^w(t)}e(t)\frac{dy(t)}{d\hat{W}(t)}\end{aligned} \tag{7.15}$$

where the estimated derivative is obtained from:

$$\begin{aligned}\frac{dy(t)}{d\hat{W}(t)} &= \frac{\partial y(t)}{\partial \hat{W}(t)} + \beta^w(t)\frac{\partial y(t)}{\partial x(t)}\frac{\partial x(t)}{\partial \hat{W}(t)} \\ &= \phi'(t)\hat{V}^T(t)x^T(t) + \beta^w(t)\hat{B}(t)\end{aligned} \tag{7.16}$$

Substituting equation (7.16) into equation (7.15) gives:

$$\hat{W}(t+1) = \hat{W}(t) + \frac{\alpha^w(t)}{\rho^w(t)} \cdot e(t) \cdot \left[\phi'(t)\hat{V}^T(t)x^T(t) + \beta^w(t)\hat{B}(t)\right] \tag{7.17}$$

where $\alpha^w(t)$ is an adaptive learning rate defined as:

$$\begin{aligned}\alpha^w(t) &= 1 \quad if \quad |e(t)| \geq \varepsilon_m^w \Bigg/ \sqrt{1 - \frac{\left\|\phi'(t)\hat{V}^T(t)x^T(t) + \beta^w(t)\hat{B}(t)\right\|^2 \Big/ h'_{\min}(t)}{\rho^w(t)}} \\ \alpha^w(t) &= 0 \quad if \quad |e(t)| < \varepsilon_m^w \Bigg/ \sqrt{1 - \frac{\left\|\phi'(t)\hat{V}^T(t)x^T(t) + \beta^w(t)\hat{B}(t)\right\|^2 \Big/ h'_{\min}(t)}{\rho^w(t)}}\end{aligned} \tag{7.18}$$

and where $\varepsilon_m^v = \max\left(\left|\tilde{\varepsilon}^v(t)\right|\right)$

and $\rho^w(t)$ is a normalisation factor such that:

$$\rho^w(t) = v\rho^w(t-1) + \max\left\{\bar{\rho}^w, \left\|\phi'(t)\hat{V}^T(t)x^T(t) + \beta^w(t)\hat{B}(t)\right\|^2 \Big/ h'_{\min}(t)\right\} \quad (7.19)$$

with $0 < v < 1$ and $0 < \bar{\rho}^w$ positive constants, and:

$$h'_{\min}(t) = \min\left\{\phi'_1(t)\ \ \phi'_2(t)\ \cdots\ \phi'_m(t)\right\} \neq 0 \quad (7.20)$$

The parameter $\beta^v(t)$ is a hybrid adaptive learning rate defined by:

$$\begin{aligned} \beta^w(t) &= 1, \quad \text{if } \underline{\hat{W}}(t)\hat{D}_w(t)\left\{\delta I + x(t)x^T(t)\right\}^{-1}x(t) \geq 0 \\ \beta^w(t) &= 0, \quad \text{if } \underline{\hat{W}}(t)\hat{D}_w(t)\left\{\delta I + x(t)x^T(t)\right\}^{-1}x(t) < 0 \end{aligned} \quad (7.21)$$

with δI a small positive constant.

Furthermore $\hat{B}(t)$ is the extended recurrent gradient of the form $\hat{B}(t) \in \mathbb{R}^{mxn}$, defined as:

$$\hat{B}(t) = \phi'(t)\hat{V}^T(t)\underline{\hat{W}}(t)\hat{D}_w(t) \quad (7.22)$$

where $\underline{\hat{W}(t)}$ is a long vector version of the hidden layer weight matrix $\hat{W}(t)$ of the form $\underline{\hat{W}(t)} \in \mathbb{R}^{1x(mxn)}$:

$$\underline{\hat{W}(t)} = \left[\hat{W}_{1,:}(t)\ \ \hat{W}_{2,:}(t)\ \cdots\ \hat{W}_{m,:}(t)\ \right] \quad (7.23)$$

Finally $\hat{D}_w(t)$ is the Jacobian matrix for the hidden layer:

$$\hat{D}_w(t) = \left[\hat{D}_w^1(t)\ \ \hat{D}_w^2(t)\ \cdots\ \hat{D}_w^m(t)\right] \quad (7.24)$$

where:

$$\begin{aligned} \hat{D}_w^i(t) &= \left[\ \frac{\partial u(t)}{\partial \hat{W}_{i,:}(t)} \cdots \frac{\partial u(t-d)}{\partial \hat{W}_{i,:}(t)}\ \frac{\partial y(t-1)}{\partial \hat{W}_{i,:}(t)} \cdots \frac{\partial y(t-l)}{\partial \hat{W}_{i,:}(t)}\ \right]^T \\ &\approx \left[\ \frac{du(t)}{d\hat{W}_{i,:}(t)} \cdots \frac{du(t-d)}{d\hat{W}_{i,:}(t)}\ \frac{dy(t-1)}{d\hat{W}_{i,:}(t-1)} \cdots \frac{dy(t-l)}{d\hat{W}_{i,:}(t-l)}\ \right]^T \end{aligned} \quad (7.25)$$

Equations (7.18) to (7.25) provide the elements required to evaluate equation (7.17) such that the hidden layer weight matrix may be updated.

The sequence of calculation using the above equations leading to weight updates is shown diagrammatically in Figure 7.2.

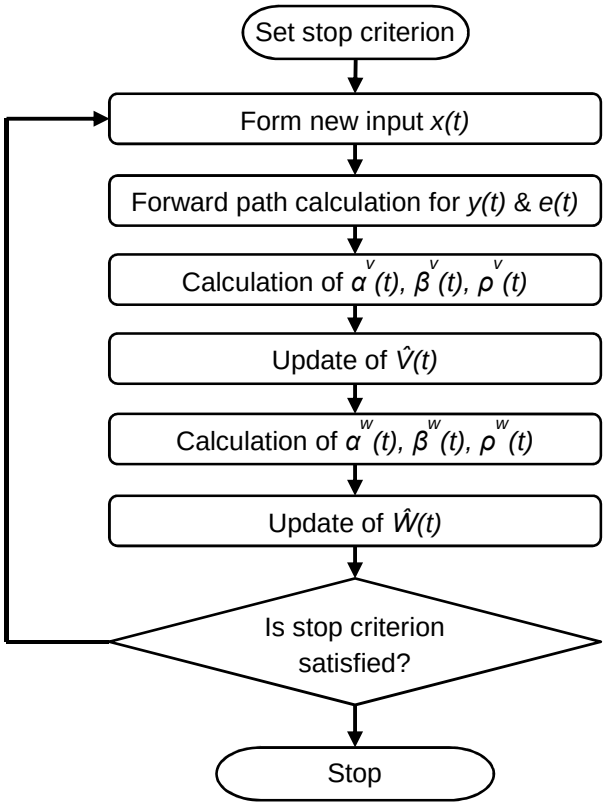


Figure 7.2 - Weight update sequence for the RAGD algorithm execution from [Song et al. 2008]

7.4 Implementation of the RAGD Training Algorithm Implementation for Cylinder Pressure Reconstruction

On initial completion of the RAGD implementation in MATLAB, first efforts to use the algorithm for training a NARX network for cylinder pressure reconstruction were disappointing. Training behaviour proved unstable, with large variations in weight values and very poor correlation of predicted data to target, even within the training data set, let alone on unseen data. Initially it was not clear whether the root cause lay in errors of implementation of the algorithm, poor values of the adjustable variables available within the method, or difficult features of the datasets.

To eliminate the latter of these issues, a simple modelled dataset was developed to ensure the data was smooth and continuous with minimum non-linearities that were not related to the cylinder pressure. To achieve this, equation (2.9) (reproduced below) with values from the engine geometry, and measured cylinder pressure (low pass filtered at 750 Hz) was used to calculate the torque due to one cylinder's gas pressure acting on the crankshaft.

$$T_g \approx P_g \pi \frac{b^2}{4} \mathrm{r} \sin\theta \left(1 + \frac{r}{l} \cos\theta \right)$$

Similar torque histories were calculated for each of the 3 cylinders (correctly phased in the angle domain), and the torque sets for each cylinder summed. This summed torque was then divided by the total inertia of the cranktrain (including flywheel, driveshaft etc.) to produce a synthesised crankshaft angular acceleration. This synthesised crank acceleration contains only information relating to cylinder pressure and crank angular position, and seeks to train a network to solve equation (7.26) to reconstruct the cylinder pressure P_g.

$$P_g = \frac{\alpha_c I_c}{\pi \frac{b^2}{4} \mathrm{r} \sin\theta \left(1 + \frac{r}{l} \cos\theta \right)} \tag{7.26}$$

where P_g is the cylinder pressure
α_c is the crankshaft angular acceleration
I_c is the cranktrain inertia
b, r, l are the cylinder bore, crank throw and con-rod length
θ is crankshaft angle from TDC

Figure 7.3 illustrates an example of the data used, showing the measured cylinder pressure used as inputs (and ultimately as NARX network target values), the crankshaft torque derived from the cylinder pressures alone, and the resulting synthesised crankshaft acceleration ultimately to be used as exogenous inputs to a NARX network. The approach intended was to use this data with a simplified information content to understand the capabilities

of the RAGD training implementation, and then introduce measured crankshaft acceleration if the method showed promise.

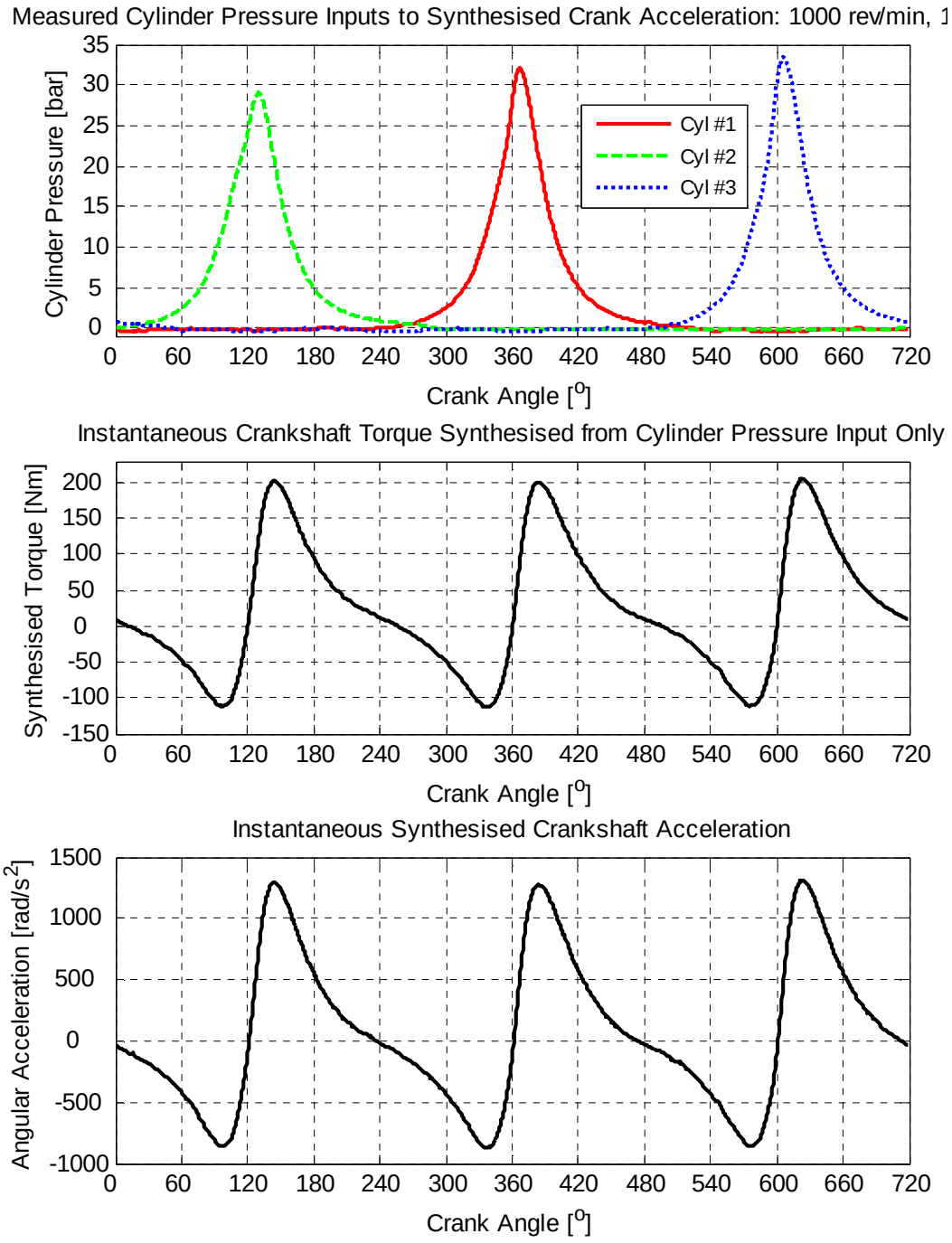


Figure 7.3 – Development of synthesised crankshaft acceleration:
Upper: Measured cylinder pressure inputs
Centre: Cylinder pressure derived torque on crankshaft
Lower: Synthesised crankshaft acceleration

Initial attempts at training with the MATLAB implementation of the RAGD method showed some promise. Figure 7.4 shows an example of an early training iteration. Normalised cylinder pressure compressed to a scale of 0 to

1 is predicted on-line by the algorithm, although some instability appears after approximately 3 engine cycles, the predicted trace follows the target well before this, and continues to recover from unstable excursions and produce a reasonable pattern of prediction around peak pressure location.

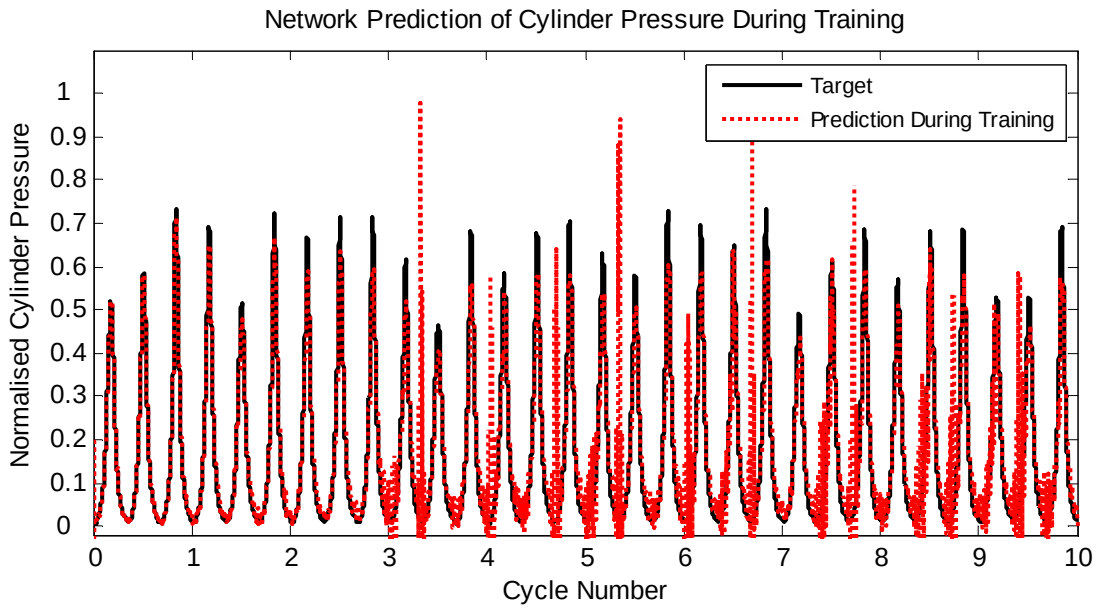


Figure 7.4 – Normalised cylinder pressure prediction during RAGD training iteration over 10 engine cycles with data from 1000 rev/min, 10 Nm

However, it became clear that due to the on-line nature of the weight updates, the reasonable predictions were not being made from a consistent set of weights. The weights were changing sufficiently quickly to be able match the target quite closely. This is demonstrated by the output layer weights shown in Figure 7.5, which can be seen to vary significantly through each peak pressure cycle. The problem here being that the eventual desired result would be a single, fixed set of weights that achieve good pressure reconstruction throughout an engine cycle. These early training iterations failed to deliver stability of weights through a cycle even when trained across the full 195 engine cycles available in a dataset. However, it is promising that the weight values did vary within reasonable bounds, without any weights deviating to very large positive or negative values.

Two probable factors became clear from these initial training efforts with the MATLAB implementation of RAGD – i) 195 engine cycles was not sufficient

data to achieve a stable training result, and ii) the rate at which the weights should be allowed to change should be more limited, such that sample by sample tracking of the target would be limited, and the weights would develop across rather than within peak pressure cycles.

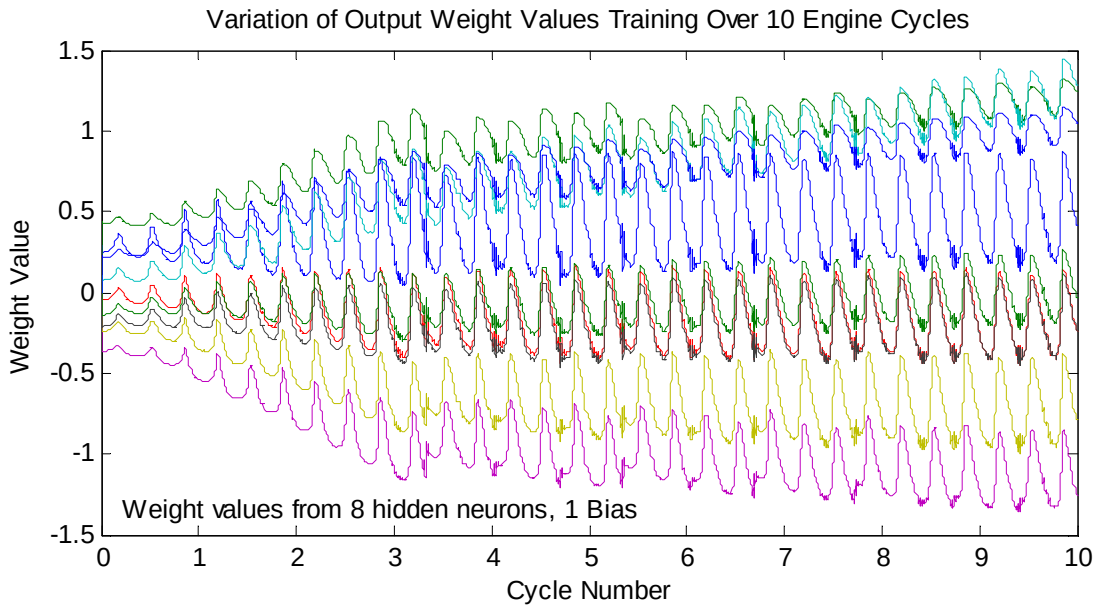


Figure 7.5 – Output weight variation during the training iteration presented in Figure 7.4

Revision to the weight update rates were achieved by changes to the constant v present in equations (7.10) and (7.19), and the constants $\bar{\rho}^{v}$ and $\bar{\rho}^{w}$ appearing respectively in the same equations. These constants influence the magnitudes of $\rho^{v}(t)$ and $\rho^{w}(t)$ which lie in the denominator of the terms governing the weight update rates in equations (7.8) and (7.17) for the output and hidden layer weight matrices respectively. Being in the denominator of the term, larger values for the constants would result in smaller update rates, and each of the constants was set to 0.9.

The problem of insufficient data to achieve a stable convergence of weight values was approached by structuring the training routine to re-use the dataset repeatedly. A set of cycles are presented to the RAGD algorithm, and the weights are updated at each sample through the signal histories. At the end of the set of cycles (an 'epoch'), the data set is re-presented to the algorithm, but now with the initial weights being the final weights from the

previous presentation. Similarly, the recurrent inputs at the start of the following epoch are also populated from the final predictions of the previous epoch, as are other RAGD parameter that require knowledge of previous time steps such as $\rho^{v}(t-1)$, $\rho^{w}(t-1)$ and recurrent terms in the Jacobian matrices.

At the end of each epoch, the mean of each weight's history over the epoch is taken, and used as a static weight set for a NARX prediction using the current test condition's unseen data set, and peak pressure error metrics calculated. No stop criteria, as such, are set but the number of epochs limited, and the prediction, error and weight histories are saved at the end of each epoch. An automated PDF report is generated at the end of each epoch with information on the RAGD parameters employed, the exogenous input vector(s), the training performance, unseen data prediction, and weight development. A summary of training and unseen data predictions for all epochs up to the current is also included. An example of a training report is available in Appendix A.

7.5 Results of the RAGD Implementation for Training a NARX Network with Synthesised Crankshaft Acceleration Inputs

Crank acceleration inputs of the type described in section 7.4 were synthesised for training data from the 1000 rev/min, 10 Nm test condition, and a series of training iterations carried out using the RAGD implementation with varying numbers of hidden neurons. These training iterations used 10 engine cycles per epoch, and training was allowed to run for 500 epochs. The figure of 10 cycles per epoch was chosen as it provided a significant amount of data per epoch (30 pressure peak events), but gave a reasonable training time per epoch to allow monitoring of progress. In each case the networks were constructed with 6 delays on the exogenous inputs (the synthesised crank acceleration), and 6 time steps of recurrent feedback from the network's output.

An overview of the progression of the training process can be gained by plotting a summary of the peak pressure metrics after each training epoch, using the mean values of P_{max} and θ_{max} at the end of each epoch, both for the training data set, and also for the unseen data set (the latter using a static weight matrices derived as the mean of the changing matrices over the epoch). Figure 7.6 shows this data for the 1000 rev/min, 10 Nm data set for a network trained with 8 hidden neurons, and Figure 7.7 shows similar data for a network training with 24 hidden neurons. Appendix B contains a full set of similar data for networks training with 4, 8 ,12, 16, 20 and 24 hidden neurons.

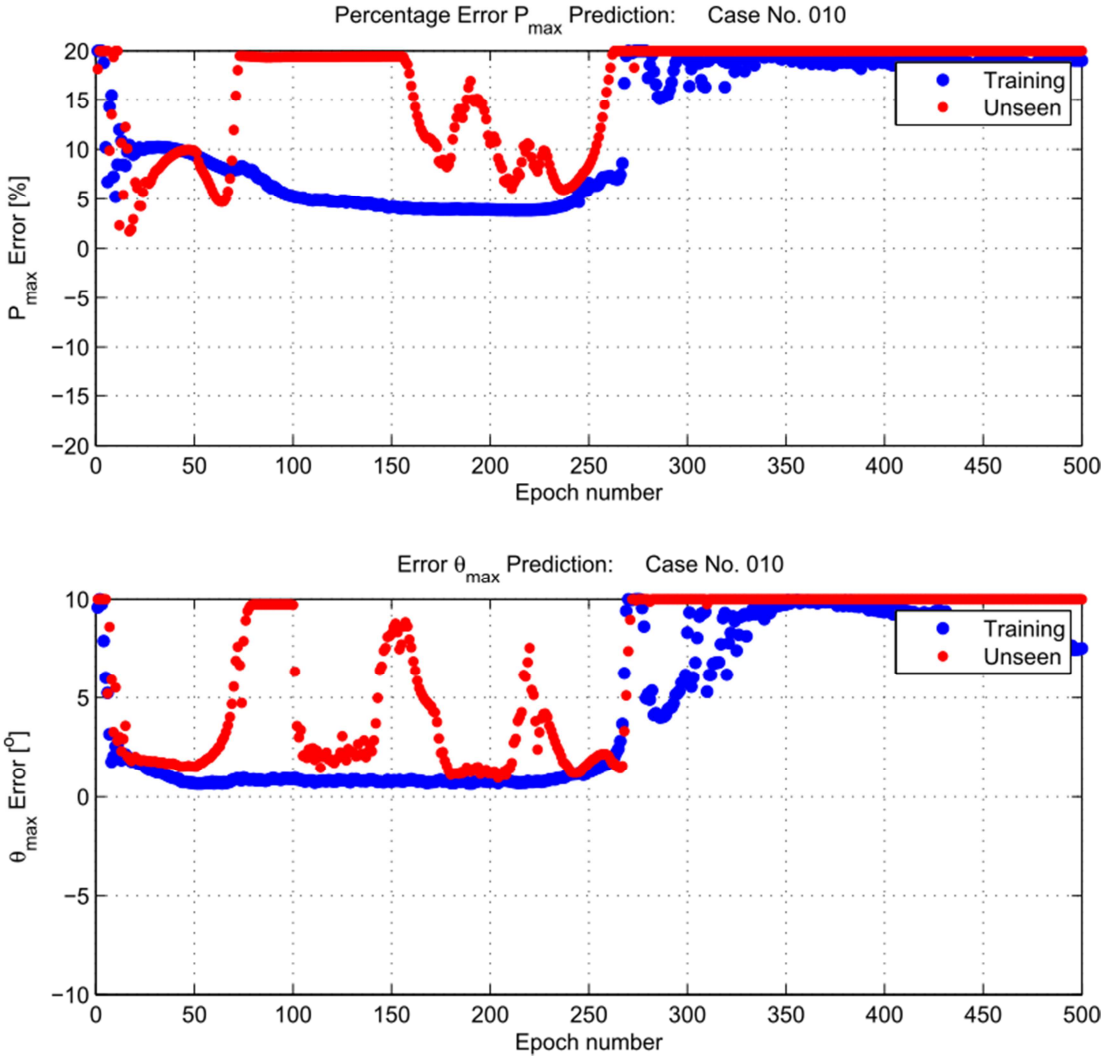


Figure 7.6 – Mean peak pressure metrics over 500 epochs, synthesised crank acceleration inputs with 8 hidden neurons 1000 rev/min, 10 Nm data sets

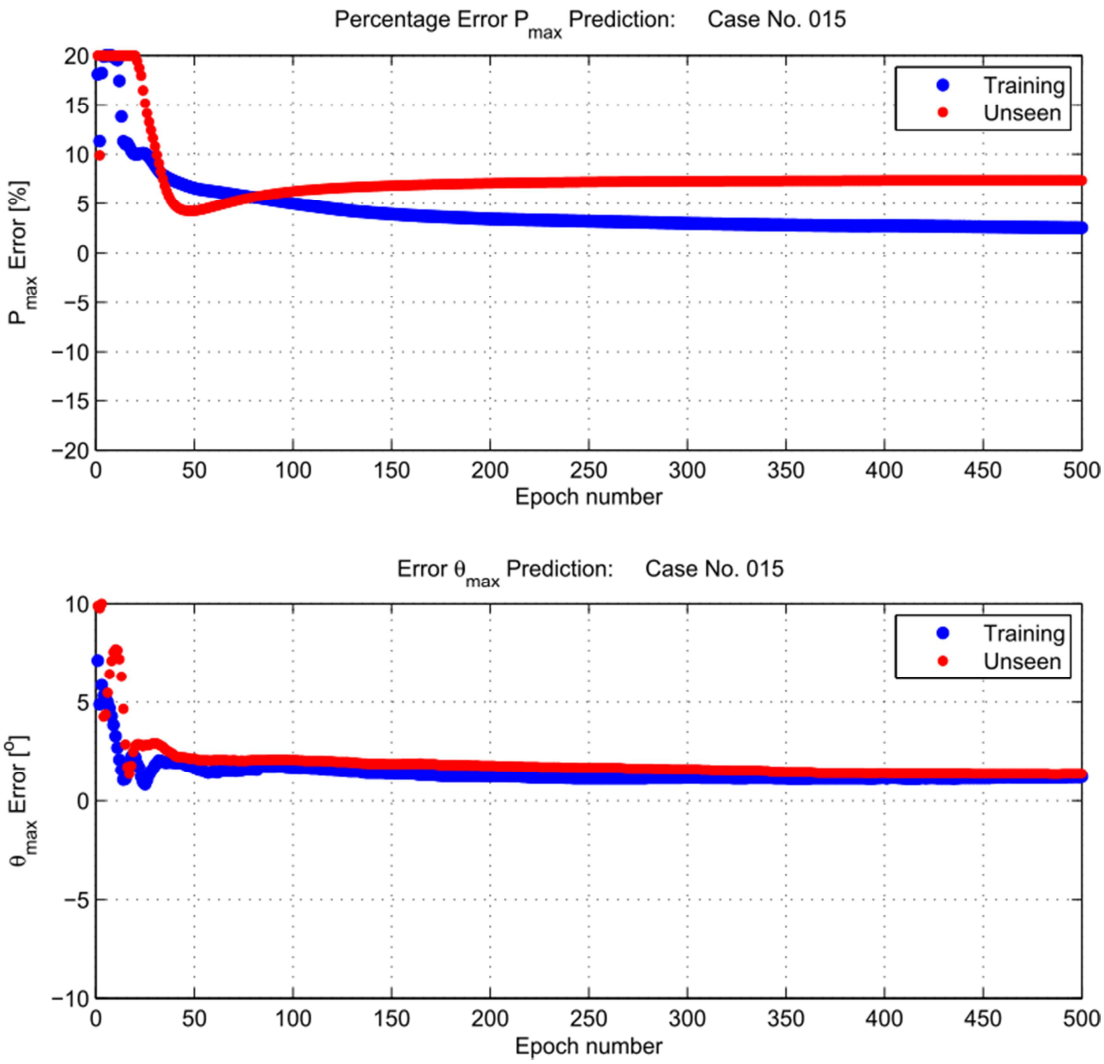


Figure 7.7 – Mean peak pressure metrics over 500 epochs, synthesised crank acceleration inputs with 24 hidden neurons 1000 rev/min, 10 Nm data sets

The data in Figure 7.6 and in Figure 7.7 suggest that the higher number of hidden layer neurons is better, the performance being more reliable over training epochs.

Inspection of Figure 7.7 suggests the best unseen data performance is achieved with network weights resulting from epochs in the range 46 to 50. The weights from epoch 47 have been recalled, and the mean of the values over the epoch used as static weights for recurrent prediction. The upper graph in Figure 7.8 shows 5 cycles of data from the training data, with network outputs (predicted cylinder pressures) overlaid on the target values. The lower graph shows similar data, but for inputs taken from the unseen data set.

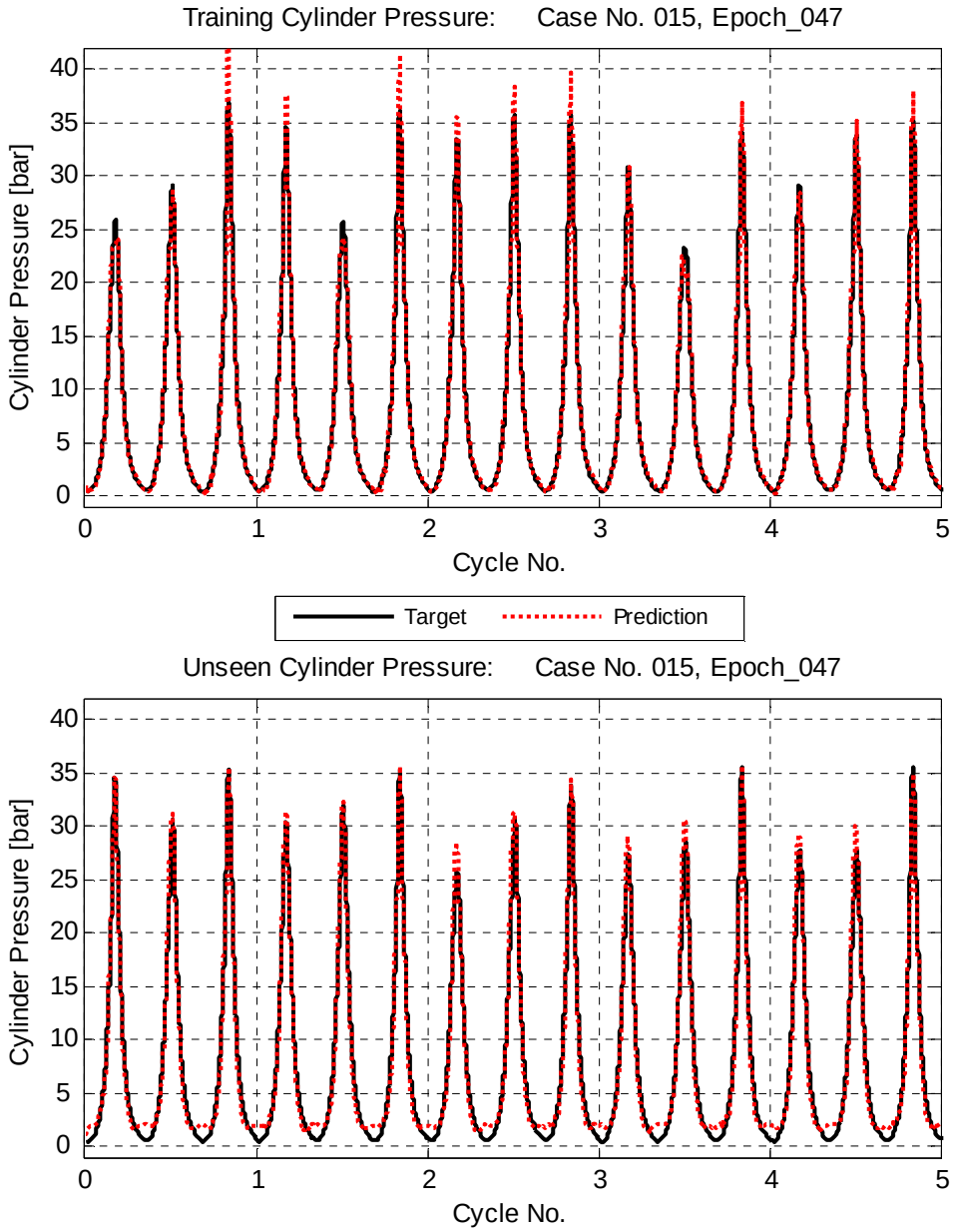


Figure 7.8 – Predicted vs. target cylinder pressure at 1000 rev/min, 10 Nm for a 24 neuron hidden layer NARX network, using data from 47th training epoch configured as static weights Upper: Prediction with the data the network was trained on Lower: Prediction with unseen data from the same test condition

The data shows a very credible pressure pattern, with peak pressure events correlating regularly with the target, and while the peak pressure amplitudes are not always close matches to the target, the pattern of higher and lower peaks is reproduced, and the network appears to capture the cylinder to cylinder, and cycle to cycle variations, rather well (qualitatively at least). Additionally, the network does not lose control of predictions at any point, and appears stable.

The full unseen data set was passed through the network producing reconstructed cylinder pressure prediction for 195 cycles. The peak pressure error metrics are show in Figure 7.9 cylinder by cylinder, with the summary statistics for all 3 cylinder combined in Table 7.1. Note that in Figure 7.9, values which exceed the Y-axis limits of the graph are shown on the limit.

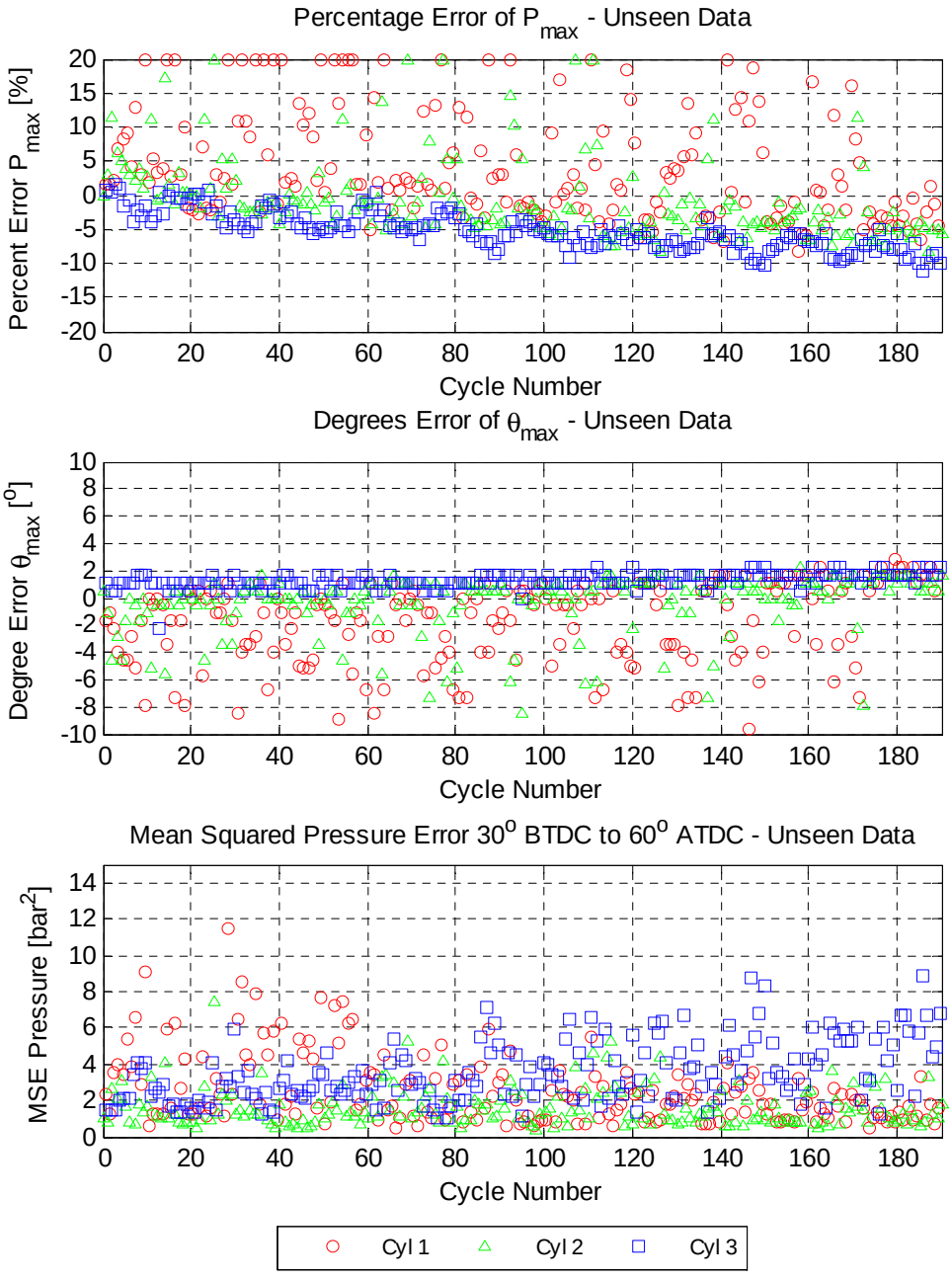


Figure 7.9 – Peak pressure metrics for RAGD prediction at 1000 rev/min, 10 Nm, weights from 47th epoch
Upper: Percentage error of magnitude of peak pressure
Centre: Degree error of position of peak pressure
Lower: Pressure MSE over window 30° BTDC to 60° ATDC

Figure 7.9 shows some mixed behaviour, cylinder 3 in particular shows low errors, albeit with some drift of the prediction across the whole dataset while cylinders 2 and 3 show somewhat greater errors.

The key summary statistics are shown Table 7.1. The mean of P_{max} values at 5.3% and θ_{max} at 1.7° are close the desired targets, though with higher than desired dispersion. Overall the results of using the implementation of the RAGD training algorithm for a NARX network trained with synthesised crankshaft acceleration derived from measured cylinder pressures shows initial promise.

Exogenous Input to ANN: Synthesised Crankshaft Acceleration	Network trained at 1000 rev/min, 10 Nm			
	Mean error P_{max} [%]	2σ Error P_{max} [%]	Mean error θ_{max} [^{o}]	2σ error θ_{max} [^{o}]
1000 rev/min, 10 Nm	5.3	9.1	1.7	3.4
1000 rev/min, 20 Nm	7.7	12.9	3.8	4.8
1500 rev/min, 10 Nm	33.6	10.3	4.3	1.4

Table 7.1 – Summary statistics of peak pressure error metrics for NARX networks predicting on unseen data

The final experiment with synthesised crankshaft acceleration inputs was to understand how a network trained at one test condition would predict data recorded at another. Further crankshaft acceleration signals were synthesised using the measured cylinder pressures from:

i) 1000 rev/min, 20 Nm (i.e. a change of load)
ii) 1500 rev/min, 10 Nm (i.e. a change of speed)

These signals were applied to the network trained at 1000 rev/min, 10 Nm, using static weights derived from the 47th training epoch. Figure 7.10 shows the resulting cylinder pressure compared to the target signal for the operating condition at the same speed (1000 rev/min) but increased load (20 Nm). The predictions are not as accurate as achieved when predicting at the same speed as trained, but show a reasonable pressure profile, and once again the pattern of high and low pressure peaks is captured.

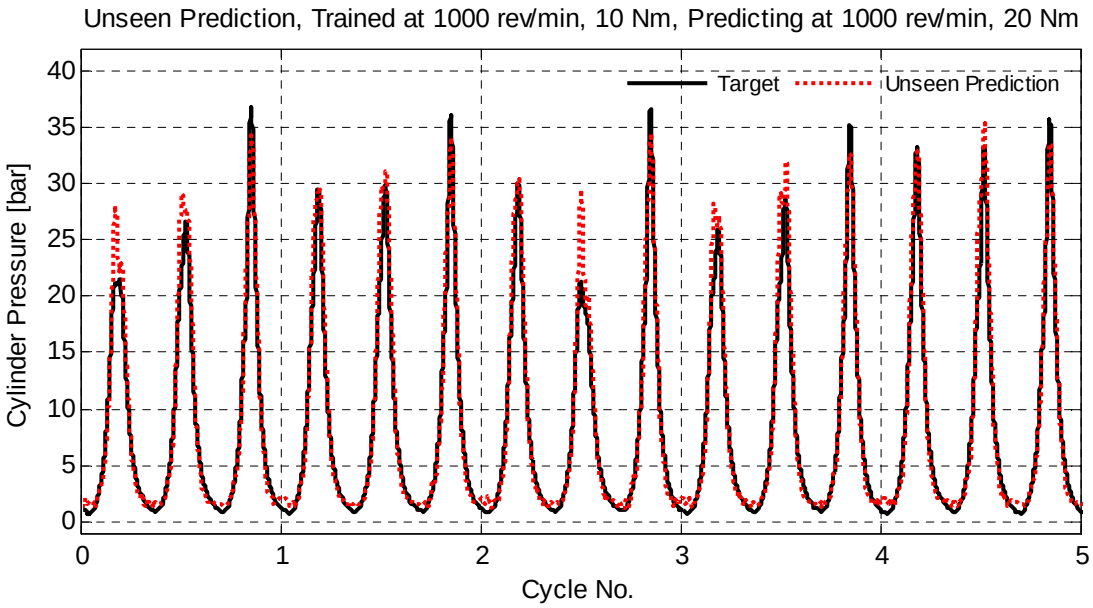


Figure 7.10 – Predicted vs. target cylinder pressure using synthesised crankshaft acceleration inputs from 1000 rev/min, 20 Nm as inputs to a network trained at 1000 rev/min, 10 Nm Solid black line: Target cylinder pressure Dashed red line: Predicted cylinder Pressure

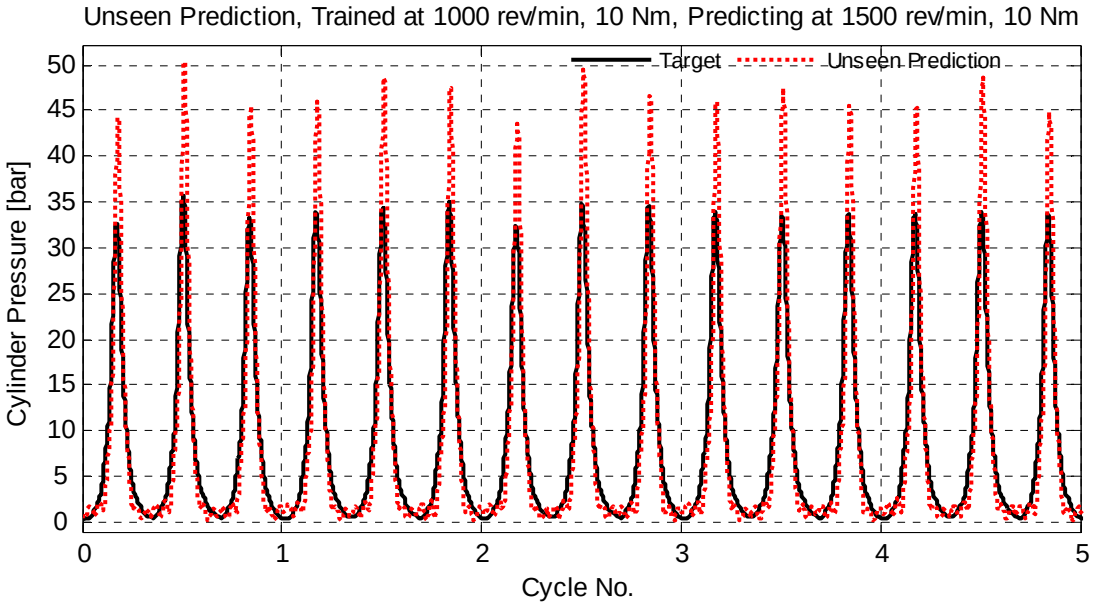


Figure 7.11 – Predicted vs. target cylinder pressure using synthesised crankshaft acceleration inputs from 1500 rev/min, 10 Nm as inputs to a network trained at 1000 rev/min, 10 Nm Solid black line: Target cylinder pressure Dashed red line: Predicted cylinder Pressure

Figure 7.11 shows similar data for the operating condition at the increased speed (1500 rev/min) but the same brake torque load (10 Nm). In this case, the amplitude predictions are significantly in error, being over predicted by more than 10 bar, however once again the pattern of high and low pressure peaks is captured.

Both conditions once again have quite stable prediction behaviour, with no permanent divergence of predicted values. Summary statistics across 195 cycles for each case are given in Table 7.1.

7.6 Results of the RAGD Implementation for Training a NARX Network with Measured Crankshaft Acceleration Inputs

Measured acceleration values were now substituted for the synthesised data described in section 7.5 above. The same network structure was used:

- 24 hidden neurons
- 6 delays on exogenous inputs
- 6 delays of recurrent feedback from network outputs

The training report across 500 epochs is shown in Figure 7.12. Training is not as successful as was seen for synthesised crankshaft acceleration, unseen data does not converge to an accurate prediction in such a stable manner. However, there are a number of epochs between the range 50-150 which appear to offer better performance than the majority.

Epoch 82 was selected from the training reports, the weights from this epoch averaged and used as static weights for NARX predictions. The upper graph of Figure 7.13 shows the overlaid target and predicted cylinder pressures for the data on which the network was trained at 1000 rev/min, 10 Nm. The lower graph in Figure 7.13 shows similar data for unseen inputs at the same operating condition. While both results are not as good as those found for the similar comparison using synthesised crank accelerations as input, they do once more produce credible pressure trace shapes and broadly capture the directional variations of peak pressure between cylinder and cycle firings.

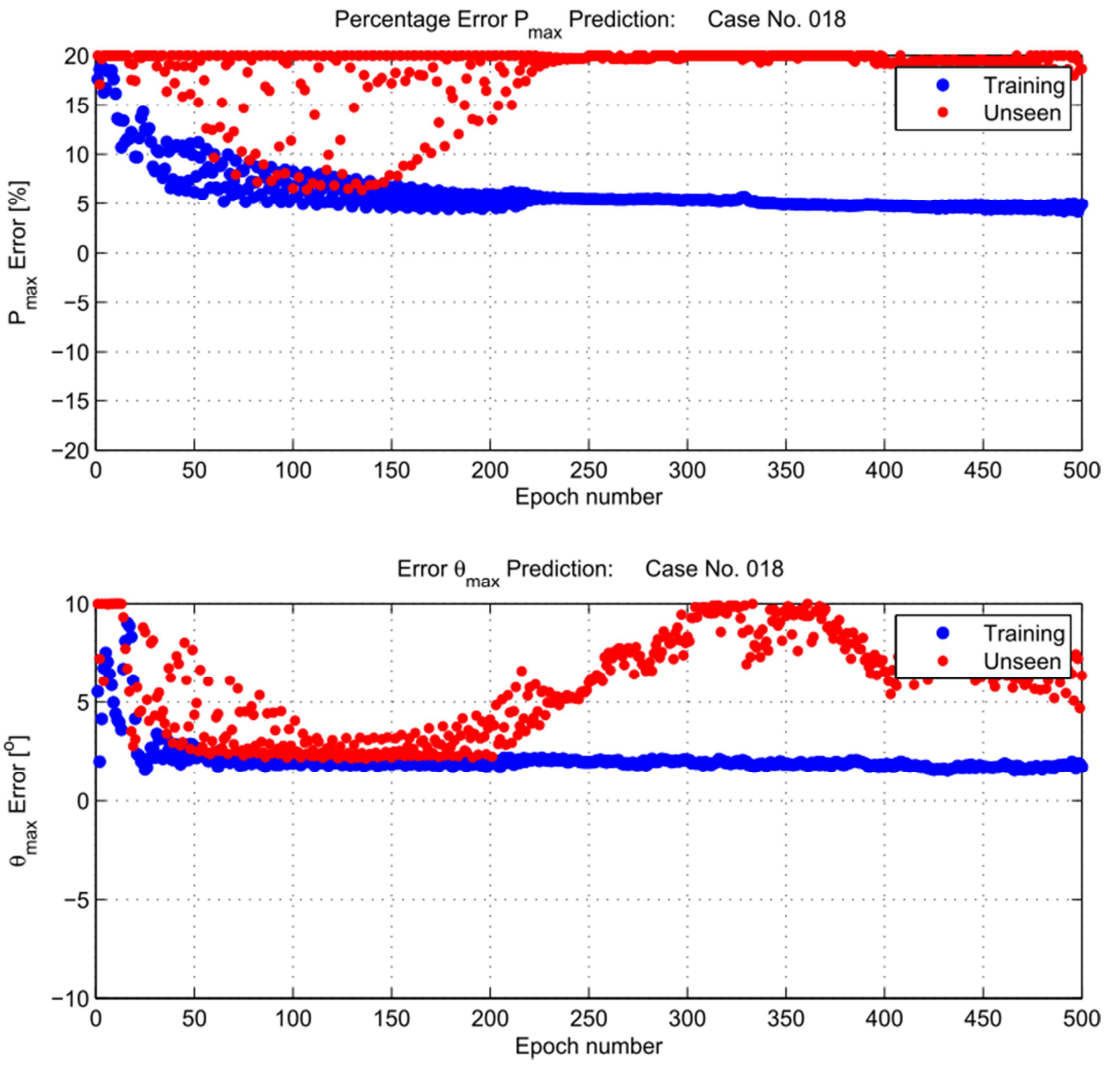


Figure 7.12 – Mean peak pressure metrics over 500 epochs, measured crank acceleration inputs with 24 hidden neurons 1000 rev/min, 10 Nm data sets

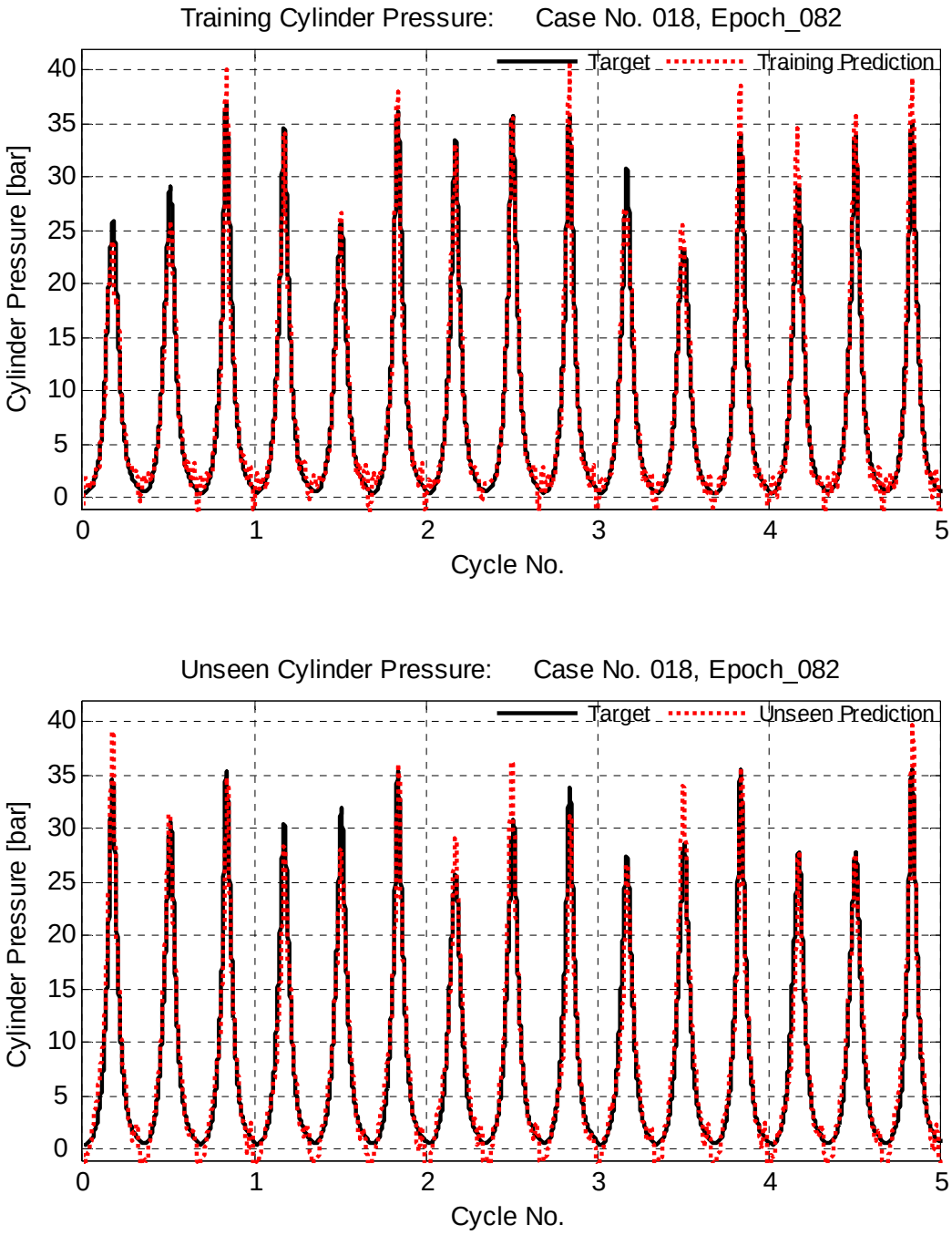


Figure 7.13 – Predicted vs. target cylinder pressure at 1000 rev/min, 10 Nm for a 24 neuron hidden layer NARX network, using measured crank acceleration
Data from 82nd training epoch configured as static weights
Upper: Prediction with the data the network was trained on
Lower: Prediction with unseen data from the same test condition

However, it seems this network does have moments of instability. Figure 7.14 shows that at some firing events later in the prediction, the predicted cylinder pressure rises to very high levels (>100bar). The network does recover well from these excursions, and returns to predicting in a reasonable

manner thereafter. The reasons for these unstable excursions are not yet understood.

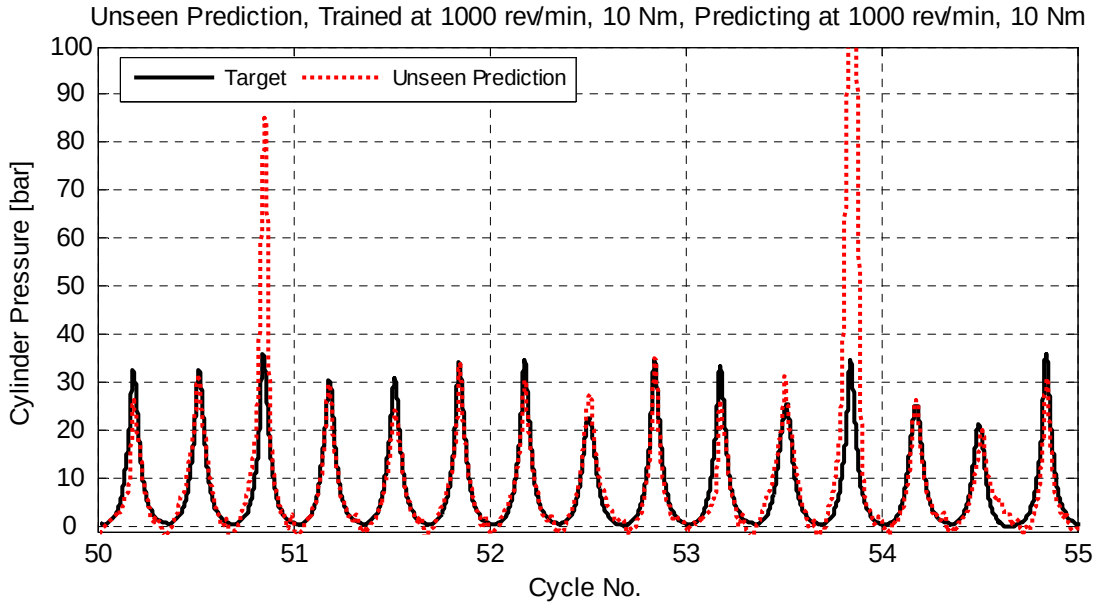


Figure 7.14 – Additional prediction on unseen data, 1000 rev/min, 10 Nm showing loss of prediction stability at some cycles

Peak pressure error metrics for the measured crankshaft acceleration input case are shown in Figure 7.15. The P_{max} data clearly shows significant dispersion from all cylinders, and the pressure peaks where prediction values lose stability can be seen by the number of markers fixed on the upper Y axis. This degree of error rather belies the relatively consistent pressure shapes seen in the time history graphs.

The θ_{max} error plot shows a more encouraging picture – although the errors significantly exceed the target band, they do remain quite well grouped in the -4° to +4° range.

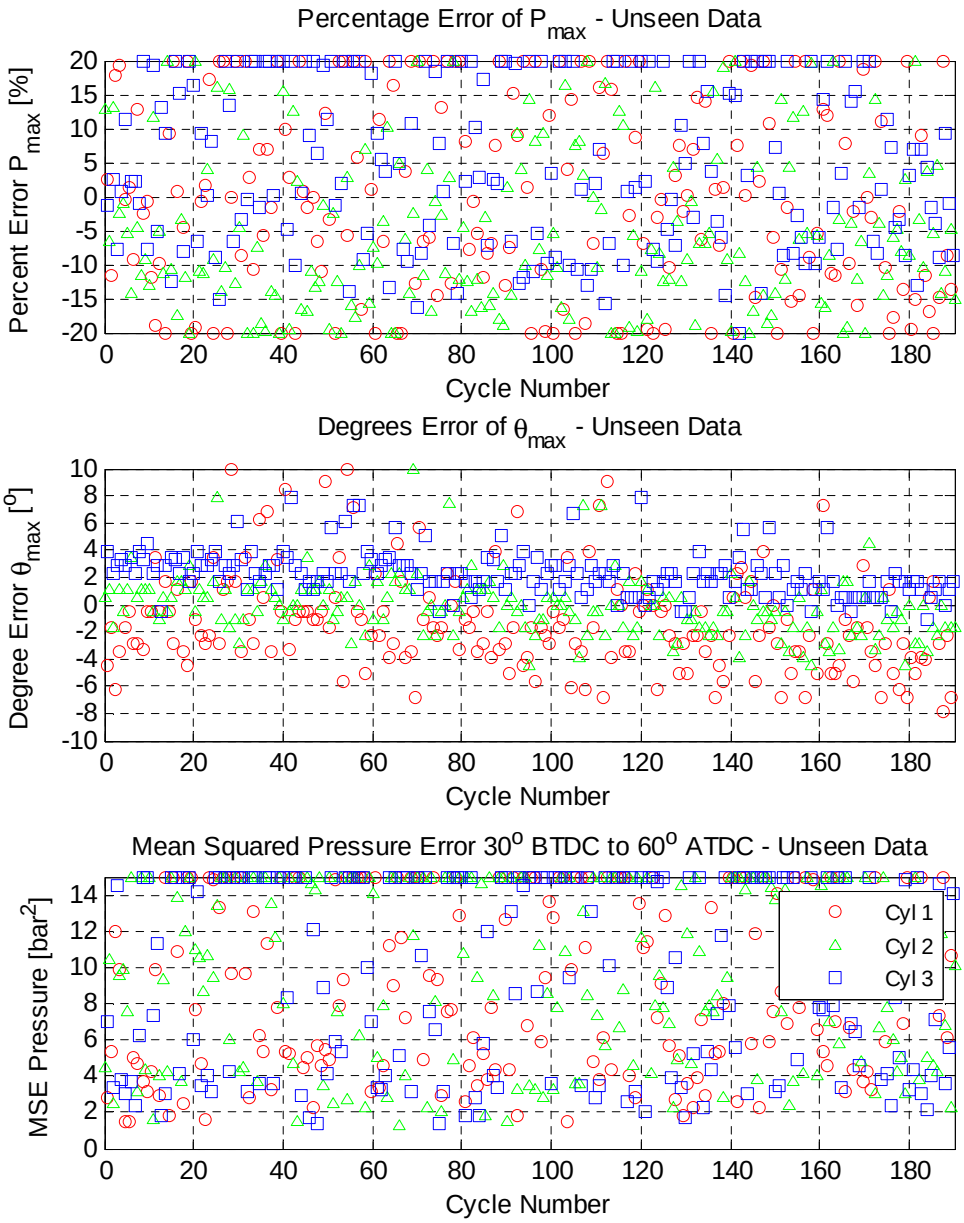


Figure 7.15 – Peak pressure metrics for RAGD prediction at 1000 rev/min, 10 Nm, weights from 82nd epoch using measured crank acceleration inputs
Upper: Percentage error of magnitude of peak pressure
Centre: Degree error of position of peak pressure
Lower: Pressure MSE over window 30° BTDC to 60° ATDC

7.7 Results of the RAGD Implementation for Training a NARX Network with Measured Crankshaft Velocity Inputs

The same network structure, and same training condition (1000 rev/min, 10 Nm) was employed to attempt network training using measured crank

shaft velocity as the input. Figure 7.16 shows the performance of the network over 100 epochs of 10 engine cycles

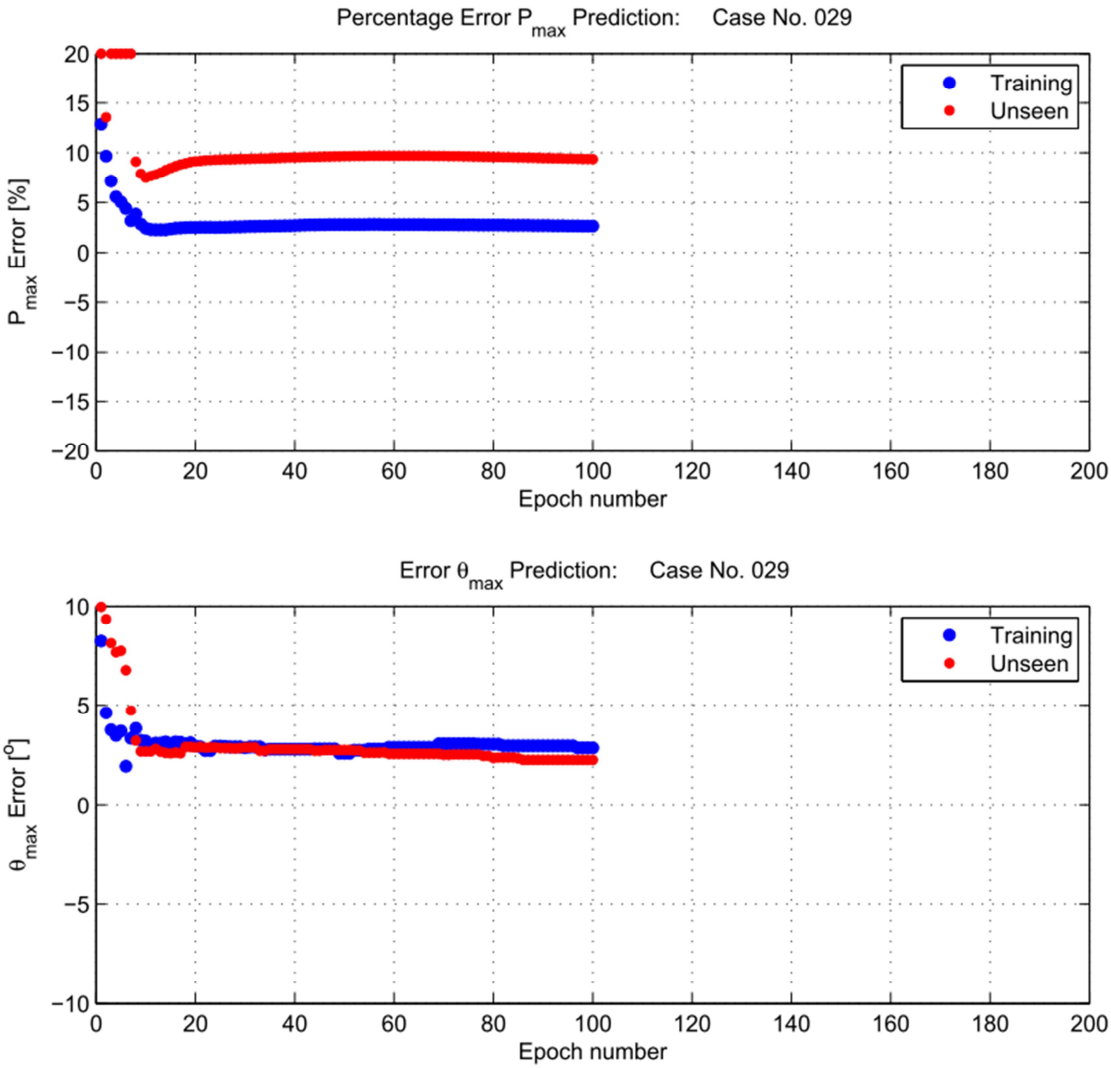


Figure 7.16 – Network performance over 100 epochs of 10 cycles while training on crankshaft velocity data

The performance develops quickly over approximately 10 epochs, but then does not improve significantly any further, however the training is much more stable than seen on measured crankshaft acceleration.

Network predictions during training (weights still changing at each time step) and unseen data (static weights) are shown in Figure 7.17. The performance is reasonable, with correctly shaped traces and the NARX prediction with fixed weights is stable with no uncontrolled excursions away from the target, but the P_{max} levels do not reach the accuracy required.

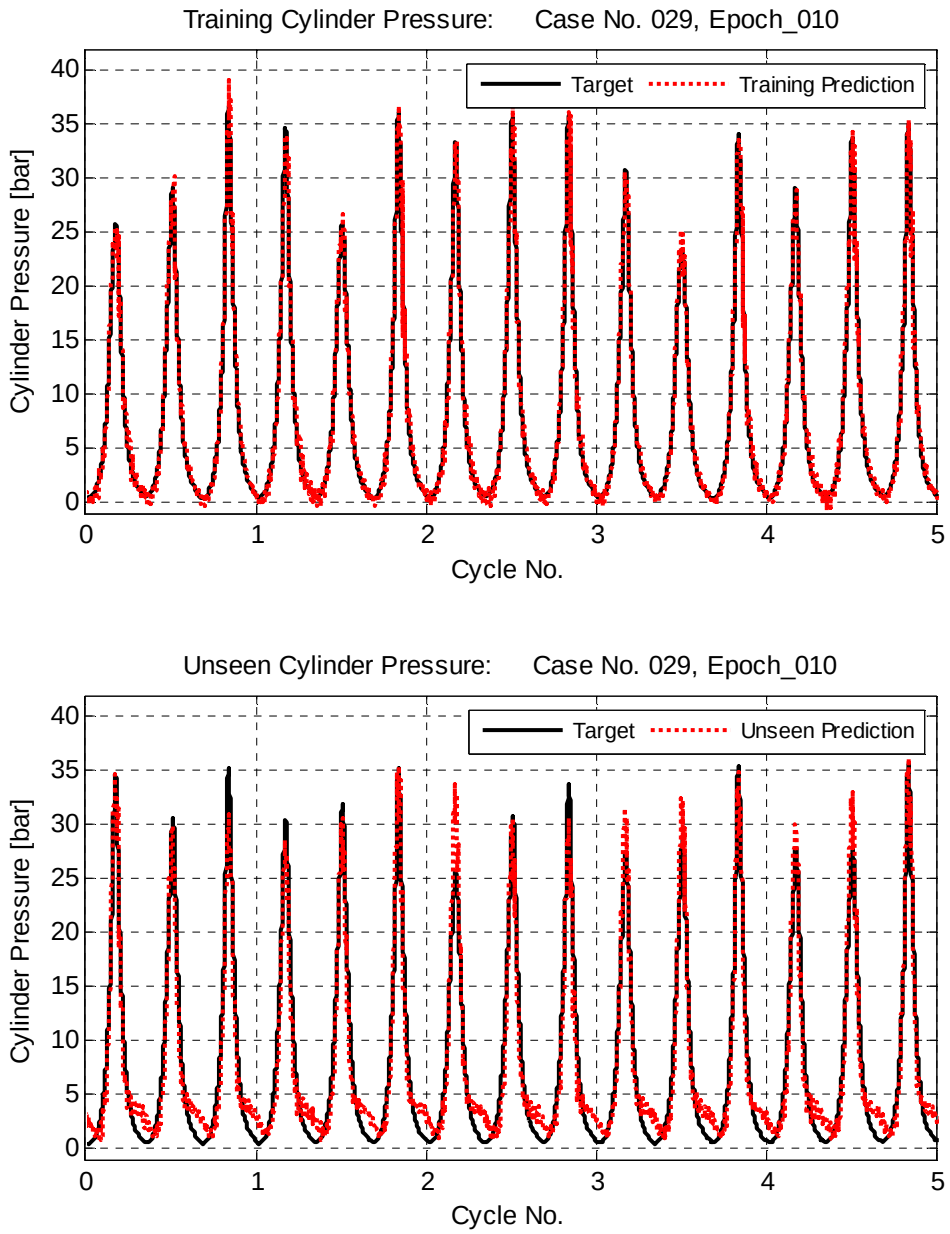


Figure 7.17 – Network performance while training on measured crankshaft velocity data at 1000 rev/min, 10 Nm
Upper: Training dataset, weights still updating
Lower: Unseen data, fixed weights

Using the static weights from training at 1000 rev/min, 10 Nm with unseen crank velocity data from 1000 rev/min, 20 Nm result in the data show in Figure 7.18. The network very significantly over predicts peak pressure in the case, though the pattern of peak pressures is matched, suggesting a need for calibration when stepping between the test conditions. Once more, the NARX prediction does remain stable. Attempting to predict an operating condition a with different speed point produced very low quality results.

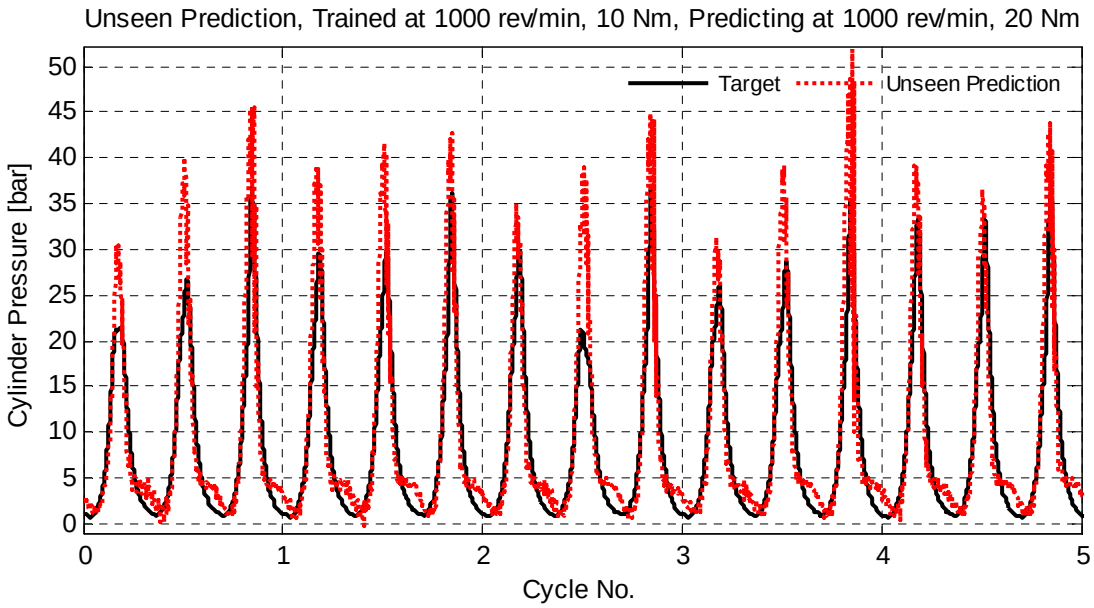


Figure 7.18 - Static weights trained at 1000 rev/min, 10 Nm, used for prediction with crankshaft velocity inputs from 1000 rev/min, 20Nm

7.8 Results of the RAGD Implementation for Training a NARX Network with Measured Crankshaft Acceleration and Measured Crankshaft Velocity Inputs

Prediction with both measures of crankshaft kinematics as exogenous inputs was tested on the same network structure as previously, but now with both exogenous inputs having 6 delayed steps. The 1000 rev/min, 10 Nm condition was once again used for training. Figure 7.19 shows the performance as the epochs advance.

The network appears to quickly achieve a reasonably good performance – within a few epochs, with the best P_{max} predictions occurring between epochs 25-30. Beyond this, performance gradually deteriorates. Figure 7.20 shows the performance on unseen data at the same condition as training using static weights from training epoch 27. The performance is similar to that for the best epoch of acceleration alone, but the training appears to be quicker in terms of epochs, and much more stable. Predictions are a little improved compared to velocity input alone.

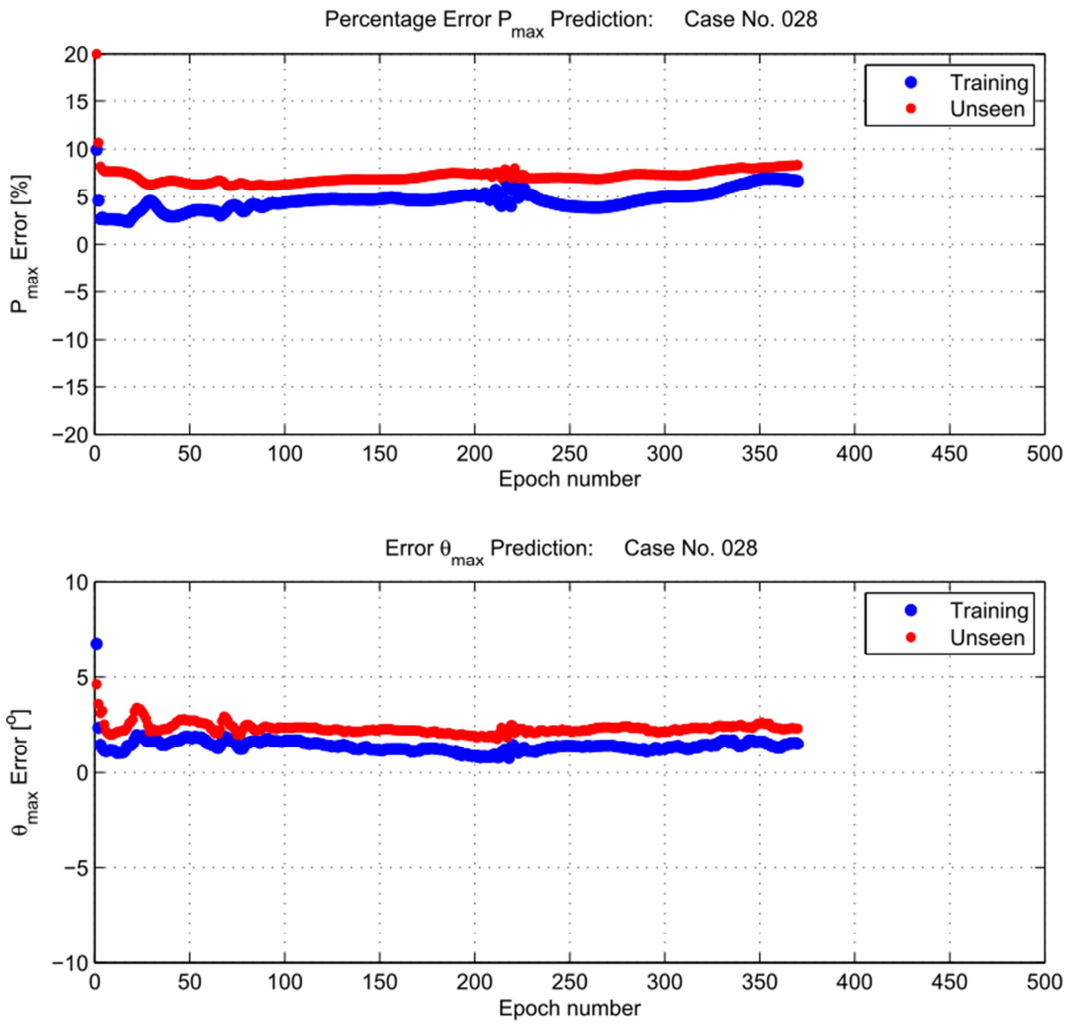


Figure 7.19– Network performance over 370 epochs of 10 cycles while training on crankshaft acceleration and crankshaft velocity data

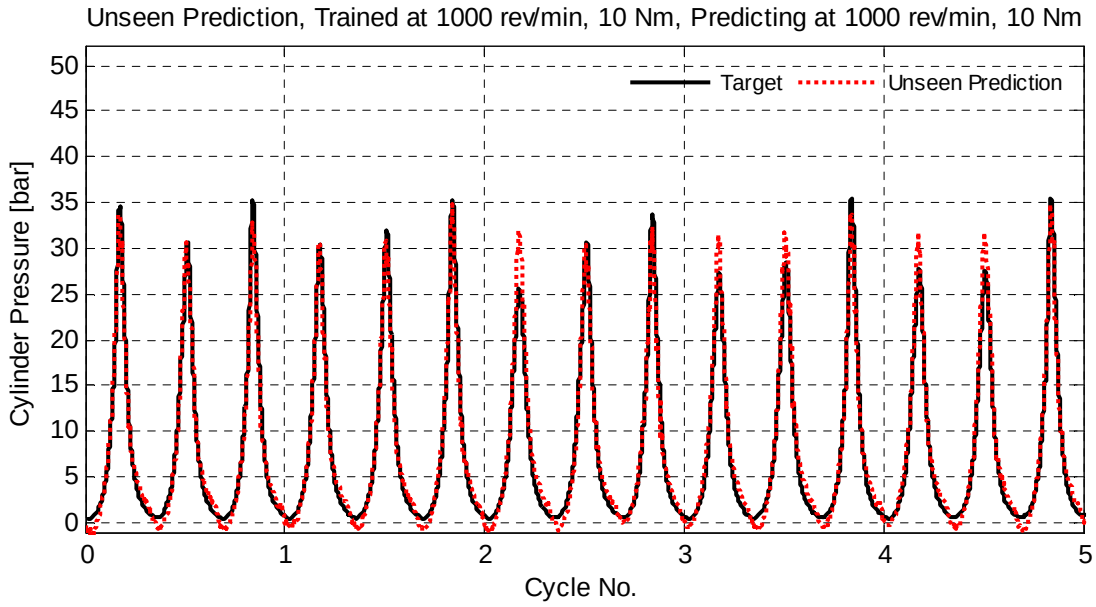


Figure 7.20 – Network performance with crank acceleration and velocity inputs on unseen data - 1000 rev/min, 10 Nm (weights from epoch 27)

Figure 7.21 shows cylinder pressure prediction metrics for the unseen data at the 1000 rev/min, 10 Nm training condition. These results are better than those seen in Figure 7.15 when measured crank acceleration alone was used to train and drive the network, but not as good as Figure 7.9 where synthesised crank acceleration was used as the input.

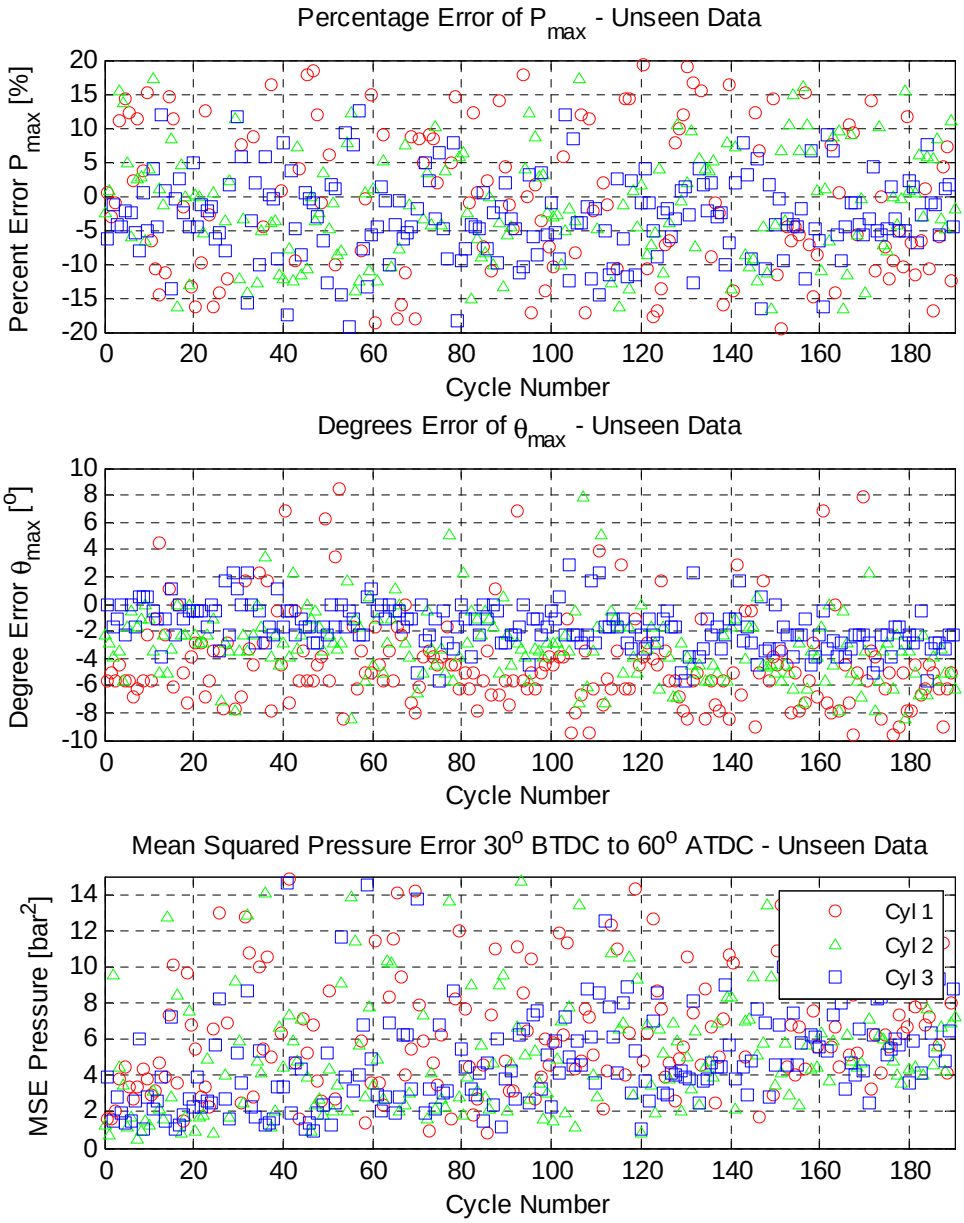


Figure 7.21 – Peak pressure metrics for RAGD prediction at 1000 rev/min, 10 Nm, weights from 27th epoch using unseen measured crank acceleration and velocity inputs Upper: Percentage error of magnitude of peak pressure Centre: Degree error of position of peak pressure Lower: Pressure MSE over window 30° BTDC to 60° ATDC

Changing load condition from 10 Nm to 20 Nm at the same training speed of 1000 rev/min gives the result show in Figure 7.22, and increasing the load further to 30 Nm, the result in Figure 7.23.

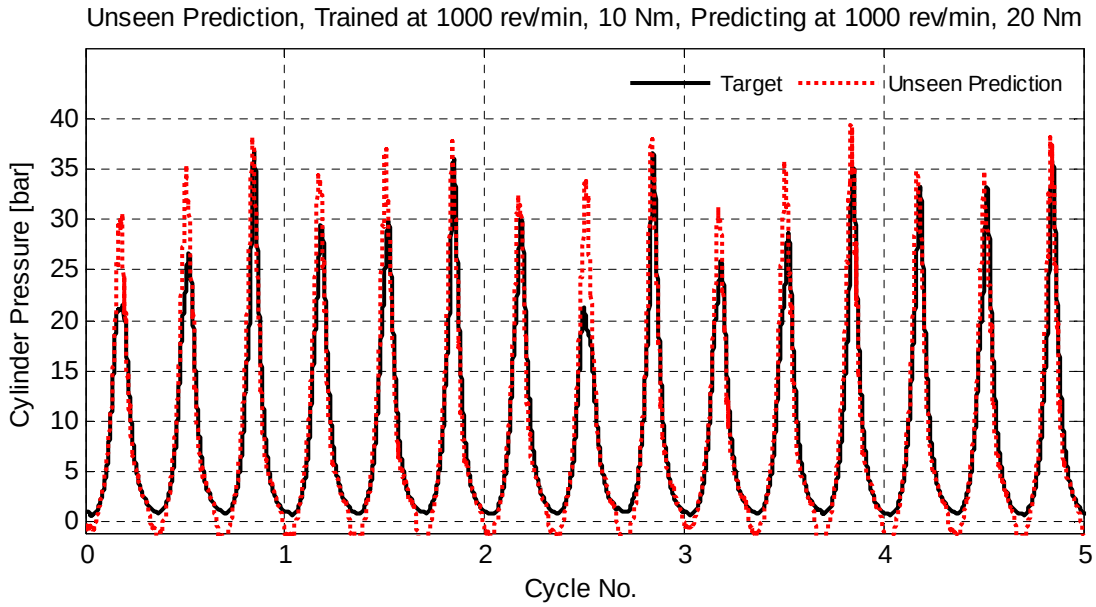


Figure 7.22 – Network prediction at 1000 rev/min, 20 Nm using weights trained at 1000 rev/min, 10 Nm

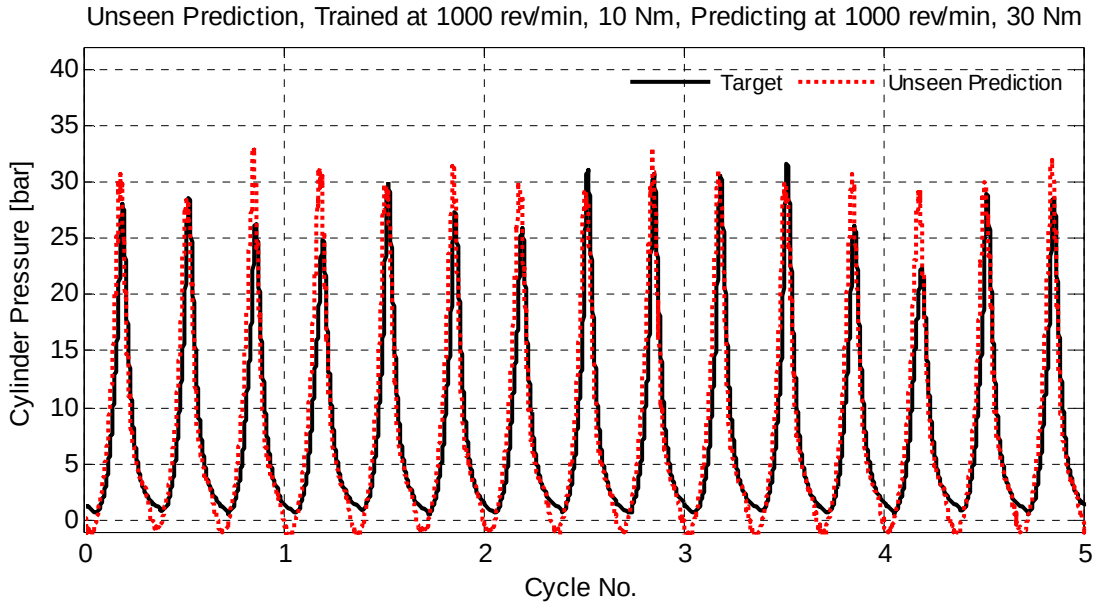


Figure 7.23– Network prediction at 1000 rev/min, 30 Nm using weights trained at 1000 rev/min, 10 Nm

Both increased load conditions have inaccurate P_{max} results, but the curve shapes are reasonable, and the predictions are stable. Making a change of speed in the test condition to 1500 rev/min, 10 Nm gives the result shown in

Figure 7.24, where it is clear the network is much less able to accommodate a change of speed than it is of load.

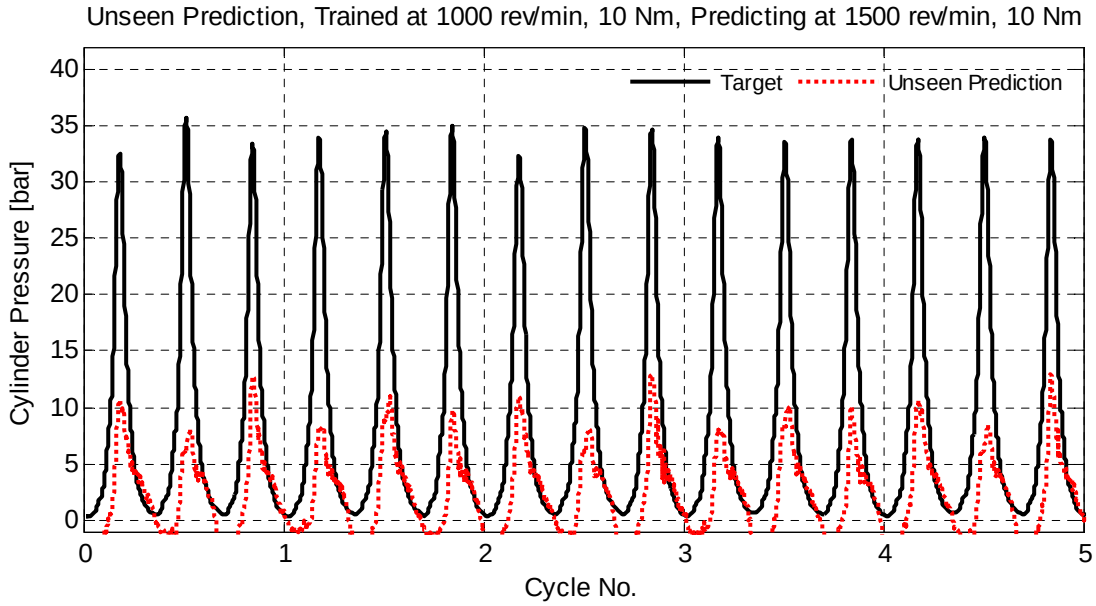


Figure 7.24 – Network prediction at 1500 rev/min, 10 Nm using weights trained at 1000 rev/min, 10 Nm

7.9 Results of the RAGD Implementation for Training a NARX Network with Measured Engine Vibration Inputs

Attempts have been undertaken to train a network on engine vibration. Data from the accelerometer was low pass filtered below 2 kHz (as identified from section 5.8). Unfortunately, this network showed no success towards useful pressure reconstruction from unseen data.

Figure 7.25 shows data during training (upper) where the weights are still adjusting within the cycle, and prediction on unseen inputs with fixed weights. The unseen prediction shows no cylinder pressure features at all, and the situation does not improve after 500 epochs. The weight changes within an epoch never reduce to the very small levels seen with crank kinematic data. Similar results were found with accelerometer data low pass filtered below 200 Hz, and with knock sensor signals low pass filtered below 2 kHz. The reasons for this poor training performance are not currently understood.

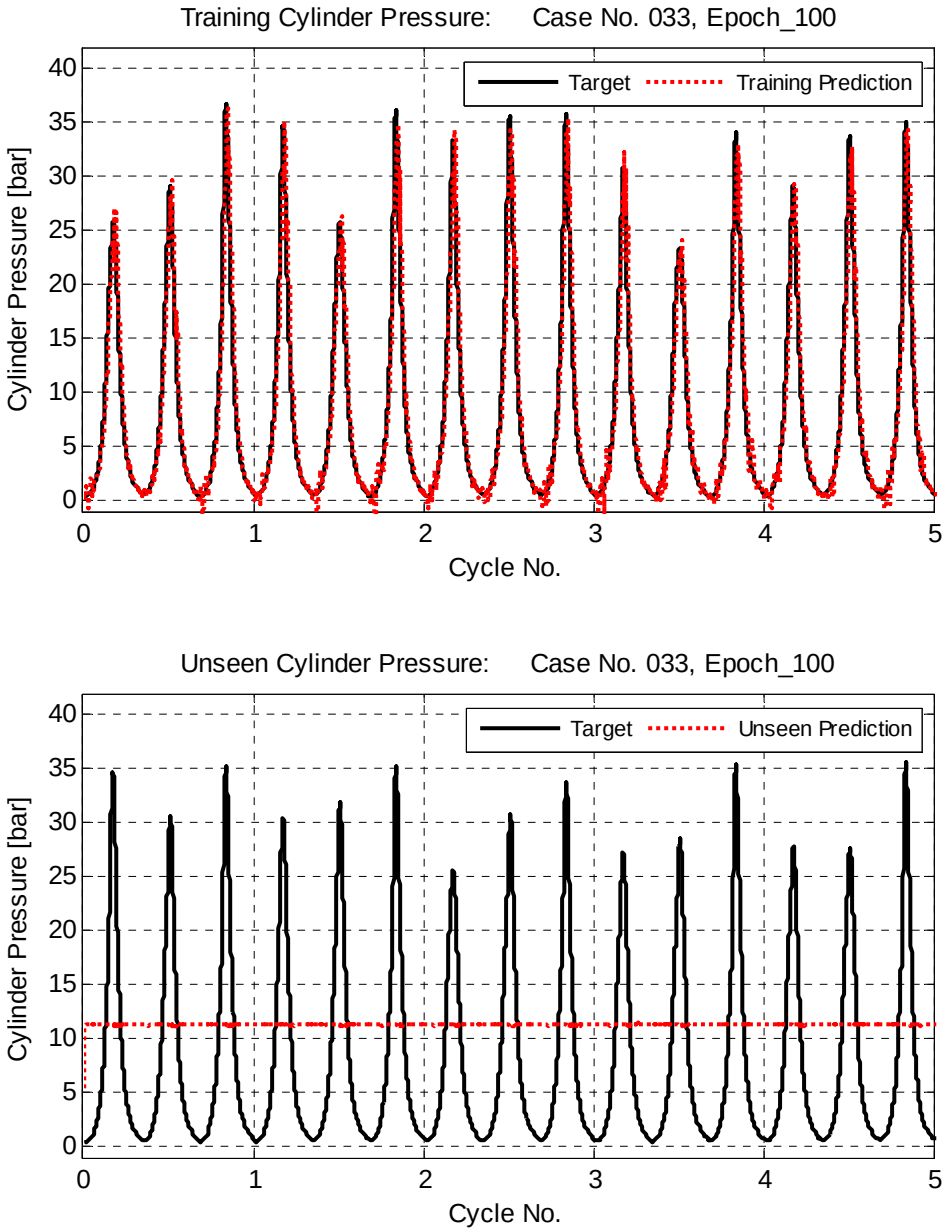


Figure 7.25 - Network trained with accelerometer signal low pass filtered <2 kHz
Upper: During training (weights still adjusting)
Lower: Unseen data predicted with static weight

7.10 Discussion of RAGD results

The RAGD recurrent training algorithm has been successfully implemented in MATLAB from the literature [Song et al. 2008]. Methods of using the algorithm have been devised to work with the signals available for input, and the extensive data manipulation and control measures around the processing have been successfully prepared.

Initial development undertaken on synthesised crank acceleration data proved the implementation of the algorithm to be sound, and defined a network structure with 6 delay terms on the input and recurrent feedback, and 24 hidden neurons.

The training method itself proves promising on measured crank kinematic data. Measured velocity input gave rather more stable training, and slightly better prediction. With combined inputs of crank angular acceleration and velocity, the network trains quite efficiently to a reasonable prediction on unseen data at the same condition as that to which it has trained. The shape of the pressure trace is credible; the NARX predictions remain stable and are capable of recovering quickly from excursions from the target. The predictive capability on data from conditions other than that on which the network was trained are significantly less accurate. Although the pressure trace shape is reasonable at the same speed but changed load, P_{max} values in particular deviate very significantly from the targets. When the speed of the input is changed, the predictions are poor.

It has not proved possible to produce good predictions from engine vibration data. During training the network weights continue to change within an epoch for much longer than seen with crank kinematic inputs, and they never converge sufficiently to result in any useful prediction on unseen data with fixed weights.

Chapter Eight

8. DISCUSSION AND CONCLUSIONS

8.1 Discussion

The need and reasons for interest in cylinder pressure reconstruction have been set out, and the majority of the objectives of this thesis have been tackled.

The first objective of this thesis was to gather a reference set of operating signals from a gasoline engine to provide high quality input data for artificial neural network training and testing towards cylinder pressure reconstruction. The set of sensors installed on the Ford I3 research engine at University of Sussex have been coupled to a digital data acquisition system and signals recorded at a series of low speed, low load operating conditions, with sufficient engine cycles for network training studies. A suite of cylinder pressure signals, crank kinematic and engine block vibration measurements have been recorded over a matrix of nine test conditions, consisting of combinations of 1000 rev/min, 1500 rev/min and 2000 rev/min alongside 10 Nm, 20 Nm and 30 Nm brake torques. All signals have been carefully considered for their frequency and noise contents prior to data acquisition to ensure the best quality of signal available was recorded. The dynamic performance of factors influencing crank dynamics have been considered, and both the test bed natural torsional frequencies and the crankshaft internal first modes have been established and shown to be well separated from the operating range in which data has been gathered, increasing the confidence that the crank kinematics available for network training have equal relationships to the cylinder pressure.

Data acquisition and post processing methodologies have been devised to acquire analogue signals in the time domain, to combine these with data from an optical encoder and fuse in to a single dataset which is stored in the time domain but capable of being transformed to the crank angle domain if required. The data is clearly and simply formatted within MATLAB and

available for use by researchers not involved with its acquisition. Crank kinematic data has had particular attention to overcome an inherent issue with the hardware, with a novel calibration method developed to correct manufacturing inaccuracies that otherwise were the root cause of excessive noise. Additionally, an affordable sensor feasible for production engines has been tested and compared with the optical encoder, showing good agreement within the speed ranges of interest to this research effort.

The governing physics of crank kinematics have been set out, and used to illustrate the challenge of reconstructing cylinder pressure where the pertinent input is only one of many influences on the crankshaft's rotation, and where the direct physical model cannot be inverted easily due to the geometry of the crank-slider system. Similar difficulties have been highlighted for the problem of inverting the block vibration response.

The interest in cylinder pressure reconstruction has been reviewed in the literature, highlighting the relative lack of success on gasoline engines and the increasing efforts towards the use of neural networks for this purpose that provide the thrust for this research. The topic of cylinder pressure reconstruction using neural networks has been explored, recent literature showing promise with feedforward networks using angle domain inputs of either crank kinematics or block vibration has been replicated with success, achieving mean prediction errors of 2.3% P_{max} and 1.2° θ_{max} for unseen crank acceleration input, and 3.2% P_{max} and 1.0° θ_{max} for unseen knock sensor signal input. These levels of prediction would meet the industrial targets of 4% P_{max} and 1° θ_{max}.

Time domain feedforward networks have also been implemented to provide a benchmark for the target topic of recurrent training, achieving mean prediction errors of 3.9% P_{max} and 1.5° θ_{max} for unseen crank acceleration input, again close to the industrial target.

Specific objectives of the research were to confirm the suitability of a NARX ANN structure for cylinder pressure reconstruction, and to identify and implement a suitable, up to date, fully recurrent training method to exploit the

architecture. Experiments with teacher forced, NARX architecture recurrent networks have confirmed both the promise of this structure, and the need to be able to reliably and robustly train on real measured data. A recently developed recurrent training algorithm, the Robust Adaptive Gradient Descent (RAGD) method [Song et al. 2008], has been reviewed and selected for use, and has been implemented in MATLAB in a manner suitable for use with the reference data from the Ford I3 engine. The relevant data handling structures and methods to efficiently feed large quantities of data to, and present data from the training algorithm have also been developed.

Initial difficulties with the implementation of stable RAGD training have been overcome, the use of crank kinematics synthesised from measured cylinder pressure inputs allowed focus on the method rather than the data, and training parameters have been identified that allowed the method to train successfully. This work helped achieve the objective of identifying parameters for the recurrent training method to achieve reliable training - a preferred structure for the network has been defined using 6 delays to the exogenous inputs and 6 delays on the recurrent feedback line. The hidden layer contained 24 neurons and the training rate constants $\rho^{v}(t)$ and $\rho^{w}(t)$ were set to 0.9. A method of training over repeated presentations of an epoch of data has also been developed, providing sufficient engine cycles for weight values to converge. Training on synthesised crank acceleration data from the 1000 rev/min, 10 Nm test point and then predicting using unseen input data from the same test point resulted in a mean P_{max} error of 5.3% and a mean θ_{max} error of 1.7°. This result is outside the industrial target, but shows promise.

The RAGD implementation also trained successfully though in a less stable fashion on measured crankshaft kinematic data at the low speed, low load operating point of 1000 rev/min 10 Nm, using the same network architecture and RAGD parameters developed on synthesised inputs. The predictions achieved were promising. The NARX structure behaved in a stable fashion predicting a credible shape of pressure curve, with no extreme excursions to the limits of the network's output space. On occasional greater diversions

from the target cylinder pressure, the network invariably recovered and returned to a reasonable predicted output. Crank velocity input resulted in more stable training, with slightly better predictions, than acceleration. Use of measured crankshaft acceleration and velocity as simultaneous, separate exogenous inputs resulted in a similar level of predictive accuracy, but saw the network train in fewer epochs, and with a much more stable behaviour. Prediction using unseen crank acceleration and velocity data from the same test condition as that used for training gave a mean P_{max} error of <7% and mean θ_{max} error of <3°. This set of inputs also gave a network capable of stable (though poorly calibrated) predictions at other load conditions at the same speed. Changing speed saw the predicted cylinder pressure diverge very significantly from the target, though still without loss of control.

Attempts to train a NARX network with the RAGD algorithm using block vibration signals as the exogenous inputs were not successful. The network weights did not achieve stable values through training epochs, and the predictions using unseen data were very poor. Further work around understanding this behaviour is required; as it is believed the best network predictions could come from the simultaneous use of both crank kinematic and block vibration as exogenous inputs.

Although the NARX network trained and predicted with stability on crank kinematic data and was a good match to the overall cylinder pressure target trace shape, the accuracy of the key metrics P_{max} and θ_{max} was still not sufficiently good to meet the needs of a production combustion control system, indeed feed forward angle domain or time domain networks provide better predictions on the data used for this thesis.

Implementation of a recent recurrent training method to be used with measured engine data is considered a significant success, and much work can still be conducted in understanding the influence of the algorithm's parameters. While a baseline configuration (successful on crank kinematics) has been found, the full range and effect of the method's parameters is yet to be mapped. There is now both data and training method implementation available for further work.

8.2 Conclusions

8.2.1 A reference set of operating engine data has been acquired from a Ford I3 gasoline engine. The data consists of in-cylinder pressures, crank rotational kinematics and cylinder block vibration signals and has been reduced to time histories suitable for training and testing ANNs.

8.2.2 Crank angle domain data input to feedforward networks has been demonstrated with mean prediction errors of 2.3% P_{max} and 1.2° θ_{max} for unseen crank acceleration input, and 3.2% P_{max} and 1.0° θ_{max} for unseen knock sensor signal input.

8.2.3 Feedforward time domain networks achieved achieving mean prediction errors of error of 3.9% P_{max} and 1.5° θ_{max} for unseen crank acceleration input.

8.2.4 NARX network structures have been shown to be capable of excellent reconstruction of cylinder pressure when trained by the teacher forcing method, but this approach is unreliable and more often produces poor results than good results. This conclusion promotes the need for a reliable training algorithm for the NARX network architecture.

8.2.5 The Robust Adaptive Gradient Descent (RAGD) training algorithm has been implemented in MATLAB and applied to fully measured signals, and also to simplified modelled data driven by measured cylinder pressure. A NARX network structure of 6 delays to the exogenous inputs and 6 delays on the recurrent feedback line with the hidden layer containing 24 sigmoid neurons has been trained using the RAGD training rate constants $\rho^{v}(t)$ and $\rho^{w}(t)$ set to 0.9.

8.2.6 Recurrent predictions on unseen data using weight values developed via RAGD training for synthesised crankshaft acceleration inputs, achieved a mean P_{max} error of 5.3% and mean θ_{max} error of 1.7°.

8.2.7 RAGD training of networks using measured crankshaft kinematic inputs was most stable using both crank acceleration and velocity inputs, with a mean P_{max} error of <7% and mean θ_{max} error of <3°.

8.2.8 Using measured cylinder block vibration as the exogenous input to a RAGD trained network was not successful.

9. REFERENCES

Albarbar A., Gu F., Ball A.D., Starr A. (2010), Acoustic monitoring of engine fuel injection based on adaptive filtering techniques. Applied Acoustics, Vol. 71, Issue 12 (2010), p1132–1141.

Al-Durra A., Fiorentini L., Canova M., Yurkovich S. (2011), A model-based estimator of engine cylinder pressure imbalance for combustion feedback control applications. American Control Conference 2011 p991-996 ISSN:0743-1619.

Antoni J., Daniere J., Guillet F., Randal R.B. (2002), Effective vibration analysis of IC engines using cyclostationarity Part ii} New results on the reconstruction of the cylinder pressures. Journal of Sound and Vibration Volume 257 Issue 5 November 2002, p839-856.

Ball K.J., Bowe M.J., Stone R.C., McFadden P.D. (2000), Torque estimation and misfire detection using block angular acceleration. SAE Technical Paper 2000-01-0560.

Beasley M., Cornwell R., Fussey P., King R., Noble A., Salamon T., Truscott A. (2006), Reducing diesel emissions dispersion by coordinated combustion feedback control. SAE Technical Paper 2006-01-0186.

Bizon K., Continillo G., Mancaruso E., Vaglieco B.M. (2011), Reconstruction of in-cylinder pressure in a diesel engine from vibration signal using a RBF neural network model. SAE Technical Paper 2011-24-0161.

Chauvin J., Corde G., Moulin P., Rouchan P. (2006), High frequency individual cylinder estimation for control of diesel engines. Oil & Gas Science and Technology - Rev. IFP, Vol. 61 (2006), No.1, p57-72.

Chauvin J., Grondin O., Nguyen E., Guillemin F. (2008), Real-time combustion parameters estimation for HCCI-diesel engine based on knock sensor measurement. Proceedings of the 17th World Congress, The International Federation of Automatic Control Seoul, Korea, July 6-11 2008 p8501-8507.

Connolly F.T., Yagle A.E. (1994), Modeling and identification of combustion pressure process in internal combustion engines. Mechanical Systems and Signal Processing, Vol. 8, Issue 1 January 1994 p1-19.

Connor J.T., Martin R.D., Atlas L.E. (1994), Recurrent neural networks and robust time series prediction, IEEE Transactions on Neural Networks. Vol. 5, No. 2, March 1994.

Du H., L Zhang L., Shi X. (2001), Reconstructing cylinder pressure from vibration signals based on radial basis function networks, Proceedings of the Institution of Mechanical Engineers Part D 2001. Vol. 215(6) p761-767.

El-Ghamry M., Steel J.A., Reuben R.L., Fog T.L. (2004), Indirect measurement of cylinder pressure from diesel engines using acoustic emission. Mechanical Systems and Signal Processing Vol.19, Issue 4 July 2005 p751-765.

Gao Y., Randall R.B. (1999), Reconstruction of diesel engine cylinder pressure using a time domain smoothing technique. Mechanical Systems and Signal Processing Vol.13, Issue 5 September 1999 p709-722.

Graglia R., Catanese A., Parisi F., Barbero S. (2011), The new general motors diesel engine management system. MTZ February 2011.

Gu F., Jacob P.J., Ball A.D. (1996), A RBF neural network model for cylinder pressure reconstruction in internal combustion engines. IEE Colloquium on Modelling and Signal Processing for Fault Diagnosis, September 1996. DOI: 10.1049/ic:19961374.

Gu F., Jacob P.J., Ball A.D. (1999), Non-parametric models in the monitoring of engine performance and condition Part 2: Non-intrusive estimation of diesel engine cylinder pressure and its use in fault detection. Proceedings of the Institution of Mechanical Engineers Part D 1999, Vol 213(2) p135-143.

Guesennec Y.G., Gyan P. (1999), A novel approach to real-time estimation of the individual cylinder combustion pressure for S.I engine control. SAE Technical Paper 1999-01-0209.

Hamedovi H., Raichle F., Bohme J.F. (2005), In-cylinder pressure reconstruction for multi-cylinder SI engine by combined processing of engine speed and one cylinder pressure. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) March 2005, Proceedings Volume 5, p677-680.

Harris M.C., Piersol A.G. (2002), Harris' shock and vibration handbook (Fifth edition). McGraw-Hill Inc. 2002 ISBN 0071370811.

Haskara I., Mianzo L. (2001), Real-time cylinder pressure and indicated torque estimation via second order sliding modes. Proceedings of the American Control Conference, 2001 Vol. 5 p3324-3328.

Haykin S. (1999), Neural networks a comprehensive foundation (Second edition). Prentice Hall International Inc. 1999 ISBN 0139083855.

He Y., Rutland C.J. (2004), Application of artificial neural networks in engine modelling. International Journal of Engine Research 2004 Vol.5 No.4 p281-296.

Heywood J.B. (1988), Internal combustion engine fundamentals (International edition). McGraw-Hill, Inc. 1988 ISBN 0071004998.

Irigoyen E., Pinzolas M. (2008), Numerical bounds to assure initial local stability of NARX multilayer perceptrons and radial basis functions. Neurocomputing Vol.72 Issues 1-3 Dec. 2008 p539-547.

Jacob P.J., Gu F., Ball A.D. (1999), Non-parametric models in the monitoring of engine performance and condition Part 1: Modelling of non-linear engine processes. Proceedings of the Institution of Mechanical Engineers Part D 1999, Vol.213(1) p73-81.

Johnsson R. (2006), Cylinder pressure reconstruction based on complex radial basis function networks from vibration and speed signals. Mechanical Systems and Signal Processing Vol. 20 Issue 8 Nov. 2006 p1923-1940.

Kolbeck A.F. (2011), Closed loop combustion control - enabler of future refined engine performance regarding power, efficiency, emissions & NVH under stringent governmental regulations. SAE Technical Paper 2011-24-0171.

Lee B., Guezennec Y.G., Rizzoni G. (2001), Estimation of cycle-resolved in-cylinder pressure and air-fuel ratio using spark plug ionization current sensing. International Journal of Engine Research 2001 Vol. 2 No. 4 p263-276.

Lee K., Yoon M., Sunwoo M. (2007), A study on pegging methods for noisy cylinder pressure signal. Control Engineering Practice 2008 Vol. 16 Issue 8 p922-929.

Liu F., Amaratunga A.J., Collings N., Soliman A. (2012), An experimental study on engine dynamics model based in-cylinder pressure estimation. SAE Technical Paper 2012-01-0896.

Mandic D.P., Hanna A.I., Razaz M. (2001), A normalized gradient descent algorithm for nonlinear adaptive filters using a gradient adaptive step size. IEEE Signal Processing Letters, Vol. 8, Issue 11, November 2001.

Massey J.A., Wagner R.M., Drallmeier J.A. (2011), Application of accelerometers for sensing combustion phasing of an advanced combustion engine. International Journal of Engine Research October 2011 Vol. 12 No. 5 p405-420.

Mufti R.A., Priest M. (2012), Effect of cylinder pressure on engine valve-train friction under motored and fired conditions. Proceedings of the Institution of Mechanical Engineers, Part J: Engineering Tribology April 2012 Vol. 226 No. 4 p306-314.

Müller R., Hart M., Krötz G., Eickhoff M., Truscott A., Noble A., Cavalloni C., Gnielka M. (2000), Combustion Pressure Based Engine Management System. SAE Technical Paper 2000-01-0928.

Murhpy B.J., Lebold M.S., Reichard K., Galie T., Byington C. (2003), Diagnostic fault detection for internal combustion engines via pressure curve reconstruction. IEEE Proceedings Aerospace Conference 2003, Vol. 7 p3239-3246.

Park S., Sunwoo M. (2003), Torque estimation of spark ignition engines via cylinder pressure measurement. Proceedings of the Institute of Mechanical Engineers Part D: Journal of Automobile Engineering September 2003 Vol. 217 p809-817.

Potenza R. (2006), Engine cylinder pressure reconstruction using neural networks and crank kinematics. Doctoral thesis, University of Sussex 2006.

Potenza R., Dunne J.F., Vulli S., Richardson D., King P. (2007), Multicylinder engine pressure reconstruction using NARX neural networks and crank kinematics. International Journal of Engine Research 2007 Vol. 8 No. 6 p499-518.

Rizzoni G. (1989), Estimate of indicated torque from crankshaft speed fluctuations: A model for the dynamics of the IC engine. IEEE Transactions on Vehicular Technology, Vol. 38, Issue 3, August 1989.

Rojas R. (1996), Neural networks: A systematic introduction, Chapter 7: The backpropagation algorithm. Springer-Verlag 1996 ISBN 3540605053.

Rubio J.J., Yu W., Ferreyra A. (2009), Neural network training with optimal bounded ellipsoid algorithm. Neural Computing & Applications September 2009 Vol. 18 Issue 6 p623-631.

Saraswati S., Chand S. (2010), Reconstruction of cylinder pressure for SI engine using recurrent neural network. Neural Computing & Applications September 2010 Vol. 19 Issue 6 p935-944.

Sellnau M., Matekunas A.F., Battiston P.A., Chang C-F., Lancaster D.R. (2000), Cylinder-pressure-based engine control using pressure-ratio-management and low-cost non-intrusive cylinder pressure sensors. SAE Technical Paper 2000-01-0932.

Sen A.K., Litak G., Taccani R., Radu R. (2008), Wavelet analysis of cycle-to-cycle pressure variations in an internal combustion engine. Chaos, Solitons & Fractals, Vol. 38, Issue 3, November 2008, Pages 886-893.

Song Q., Wu Y., Soh Y.C. (2008), Robust adaptive gradient descent training algorithm for recurrent neural networks in discrete time domain. IEEE Transaction on Neural Networks November 2008 Vol. 19, No.11, p1841-1853.

Stone R. (1999), Introduction to internal combustion engines (Third edition). Macmillan Press Ltd. 1999 ISBN 0333740130.

Stotsky A. (2008), Fast algorithm for real-time torque estimation. International Journal of Engine Research June 2008 Vol. 9 No. 3 p239-247.

Taglialatela F., Lavorgna M., Mancaruso E., Vaglieco B.M. (2013), Determination of combustion parameters using engine crankshaft speed. Mechanical Systems and Signal Processing July 2013 Vol. 38 Issue 2 p628-633.

Tuma J. (2002), Phase demodulation in angular vibration measurements. Proceedings of International Carpathian Control Conference May 2002 p99-104.

Villarino R., Böhme J.F. (2003), Fast in-cylinder pressure reconstruction from structure-borne sound using the EM algorithm. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) April 2003 Vol. 6 p597-600.

Villarino R., Böhme J.F. (2004), Pressure reconstruction and misfire detection from multichannel structure-borne sound. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2004 Vol. 2 p141-144.

Vulli S. (2006), Engine cylinder pressure reconstruction using neural networks and knock sensor measurements. Doctoral thesis, University of Sussex, 2006.

Vulli S., Dunne J.F., Potenza R., Richardson D., King P. (2008), Time-frequency analysis of single-point engine-block vibration measurements for multiple excitation-event identification. Journal of Sound and Vibration 321 (2009) p1129-1143.

Wagner M., Carstens-Behrens S., Johann F., Böhme J.F. (1999), In-cylinder pressure estimation using structural vibration measurements of spark ignition engines. Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics, 1999 p174-177.

Williams R.J., Zipser D. (1989), A learning algorithm for continually running fully recurrent neural networks. Neural Computation 1989 Vol. 1 p270-280.

Wu J-D., Chan J-C. (2006), Continuous wavelet transform technique for fault diagnosis of internal combustion engines. NDT&E International 39 (2006) p304-311.

Wu Y., Yang X., Song Q. (2008), A new supervised learning algorithm of recurrent neural networks and L2 stability analysis in discrete time domain. Recurrent Neural Networks, Edited by Hu X., Balasubramaniam P. Chapter 9, p169-202. InTech 2008 ISBN 9789537619084.

Yong X., Guiyou H., Chunrong S., Zhibing N., Wu Z. (2010), Reconstruction of cylinder pressure of I.C. engine based on neural networks. First International Conference on Pervasive Computing, Signal Processing and Applications, September 2010 p924-927.

Yong Cheng Y., Tang J., Ji S., Huang M. (2012), Identification of combustion timing based on the vibration acceleration of diesel engines. Proceedings of the Institution of Mechanical Engineers Part D: Journal of Automobile Engineering November 2012 Vol. 226 No. 11 p1495-1506.

Yoon P., Park S., Sunwoo M., Ohm I., Yoon K.J. (2000), Closed-loop control of spark advance and air-fuel ratio in SI engines using cylinder pressure. SAE Technical Paper 2000-01-0933.

Yu W., Rubio J.J. (2009), Recurrent neural networks training with stable bounding ellipsoid algorithm, IEEE Transactions On Neural Networks. June 2009, Vol. 20, No. 6, p983-991.

Zho J-D., Ouyang G-Y., Gao H-B. (2009), Demodulation of instantaneous rotational speed of diesel engine based on Hilbert transform. 9th International Conference on Measurements and Increments (ICEMI) 2009 Vol. 2 p753-756.

New GM V6 diesel has cylinder-pressure monitoring (2007), SAE tech briefs, April 2007.

Appendix A

Example epoch training report for RAGD implementation

Training Summary: Case No. 015, Epoch_500

RAGD Training Case No. 015 **10-Dec-2012 08:45:23**
Epoch No. 500

Data Set
Engine Speed ....: 1000 rev/min
Engine load .....: 10 Nm
Low pass filter .: 300 Hz

Network Structure
No. hidden Neurons ..: 24
Activation lam/lin ..: 1/1
Recurrent feedbacks .: 6
WgtSrc .....: 1

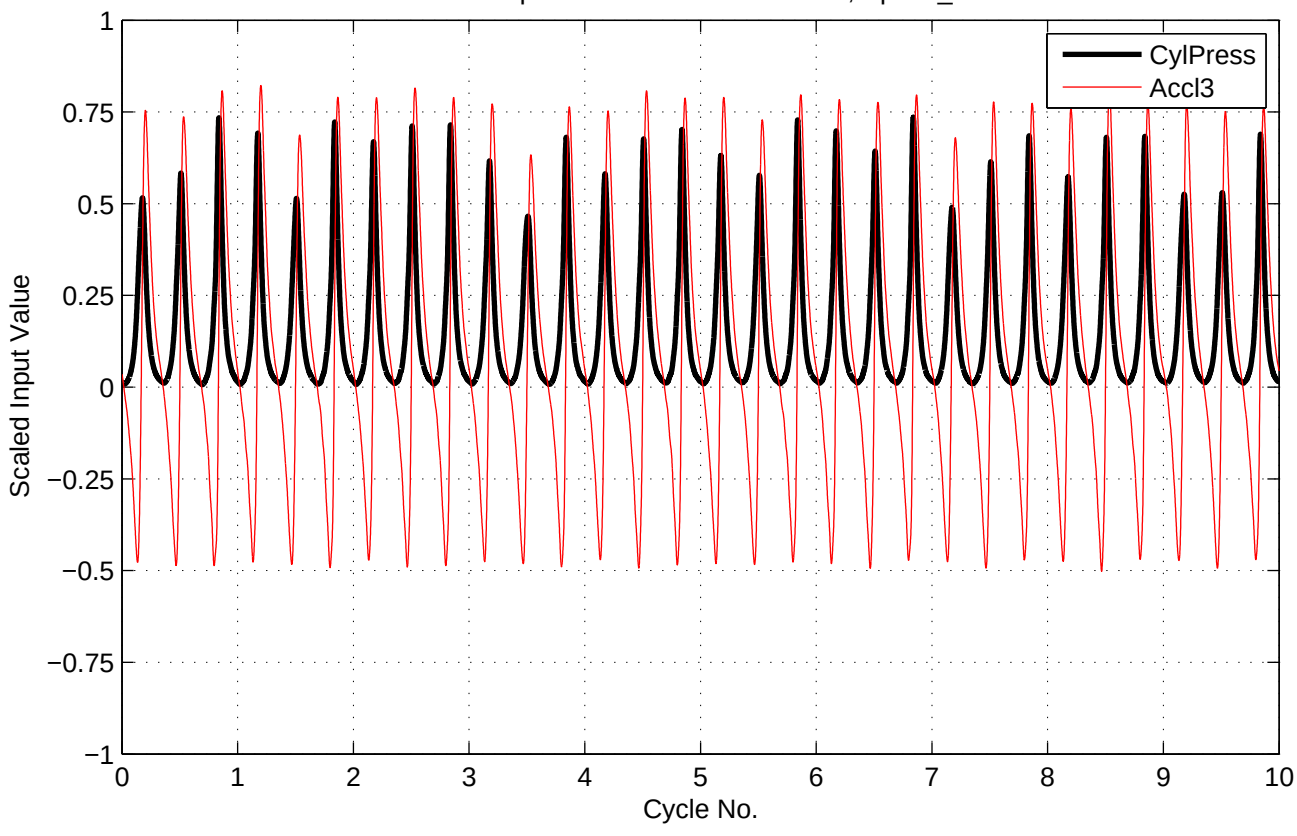
Inputs
Input 1 : Accl3 Dly : 6

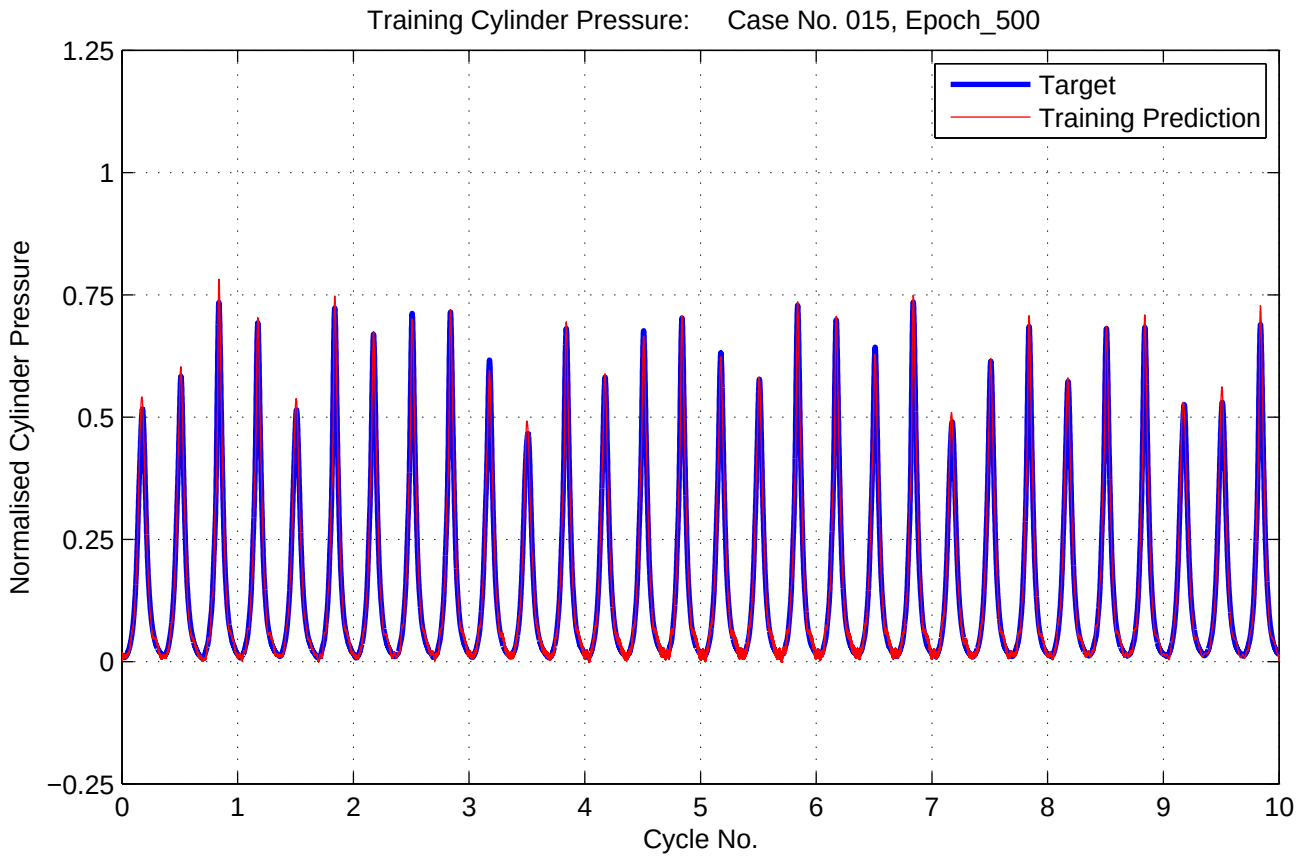
RAGD Parameters
$\bar{\rho}^v$ ..: 0.9 ρ_i^v ..: 0.5
$\bar{\rho}^w$ ..: 0.9 ρ_i^w ..: 0.5

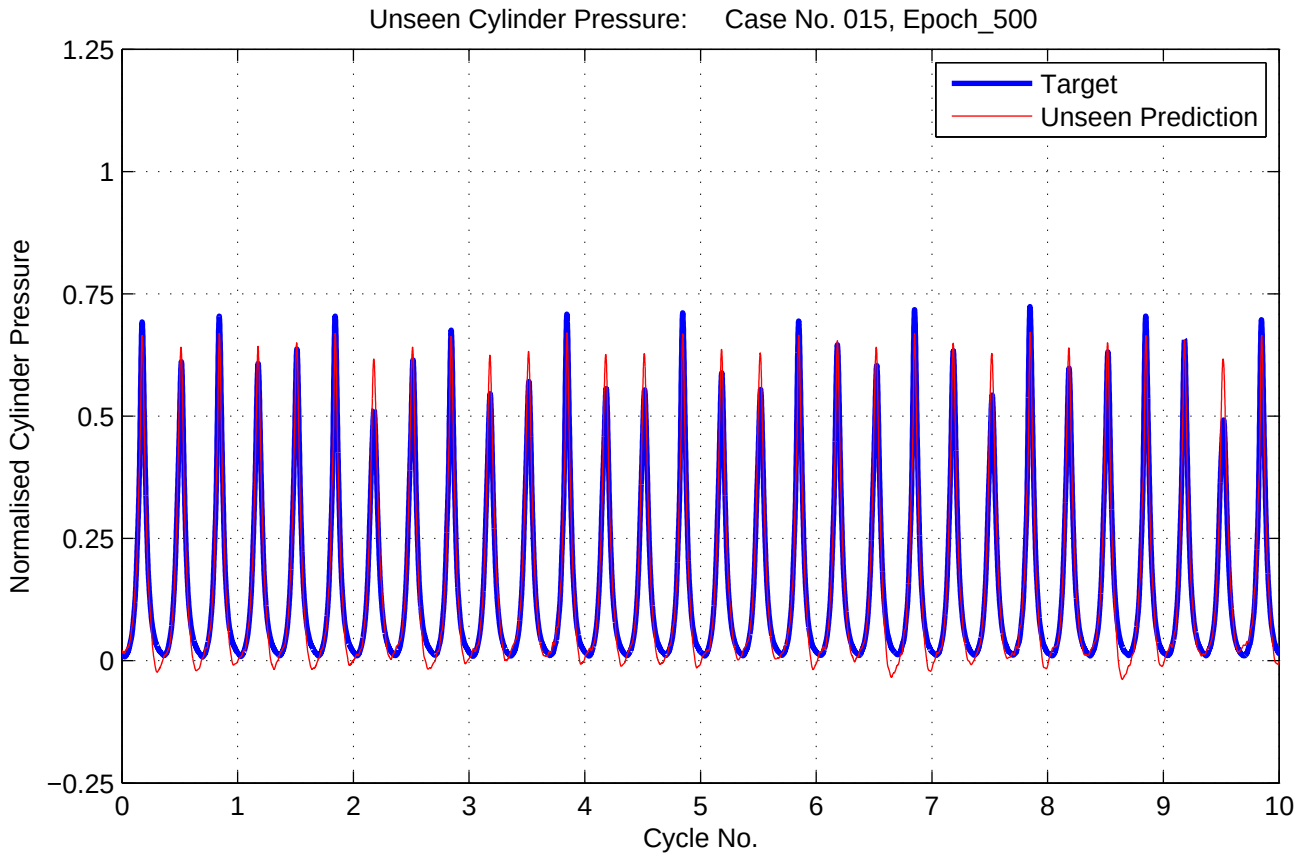
v ...: 0.9
δ ...: 0.01
ε_m^v ..: 0.00054767

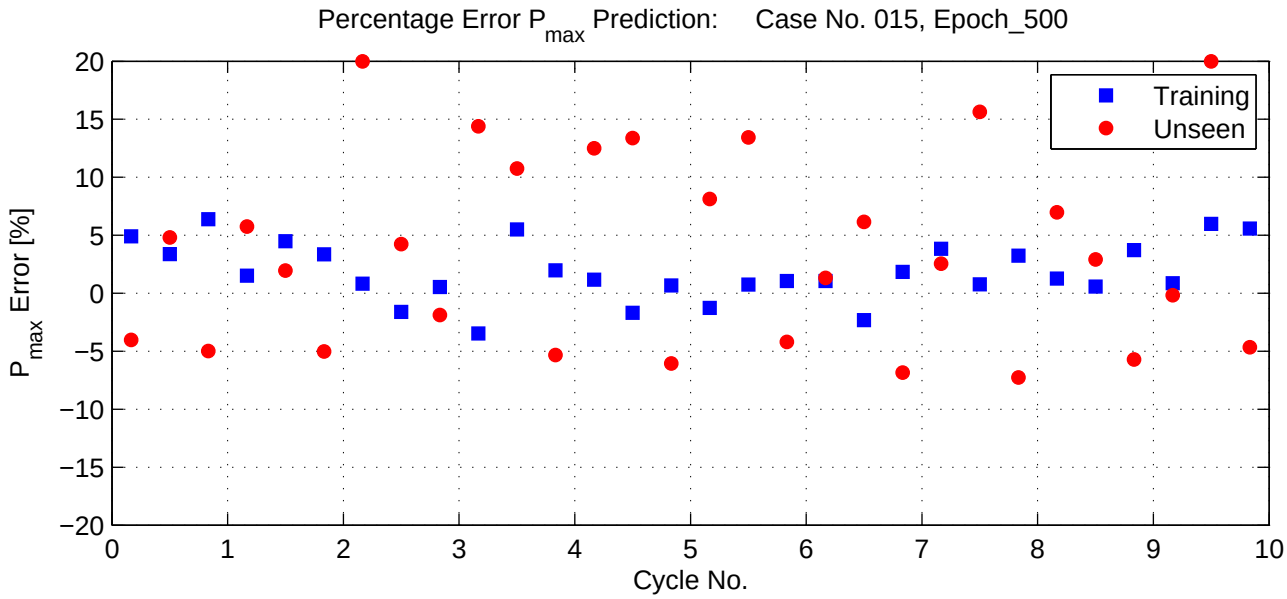
vVar ...: 0.001
wVar ...: 0.001

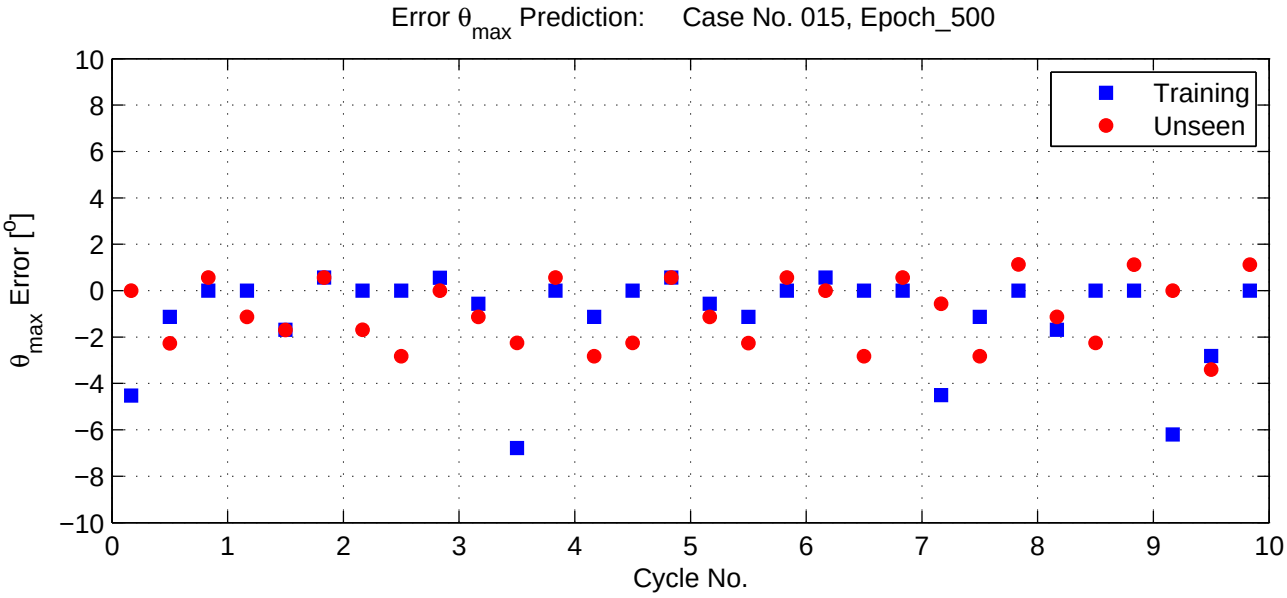
Scaled Input Vectors: Case No. 015, Epoch_500

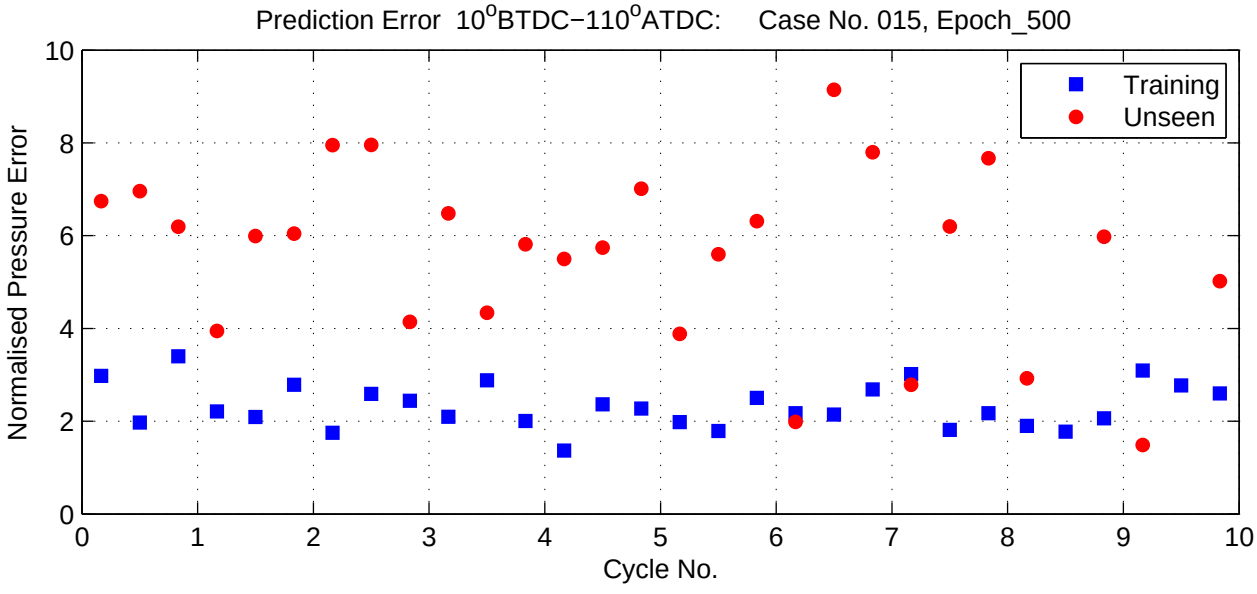


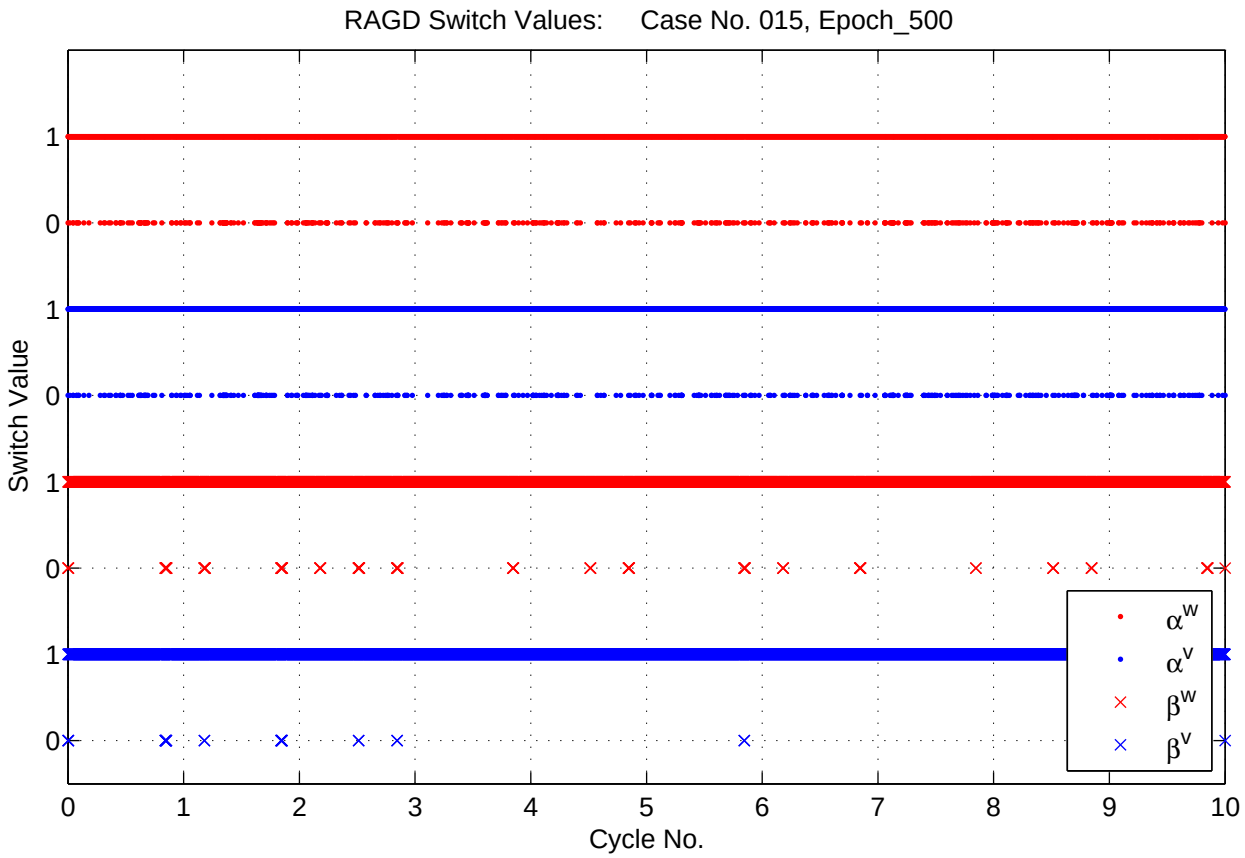

Training Cylinder Pressure: Case No. 015, Epoch_500
Target
Training Prediction
Normalised Cylinder Pressure
Cycle No.
1.25
1
0.75
0.5
0.25
0
−0.25
0
1
2
3
4
5
6
7
8
9
10

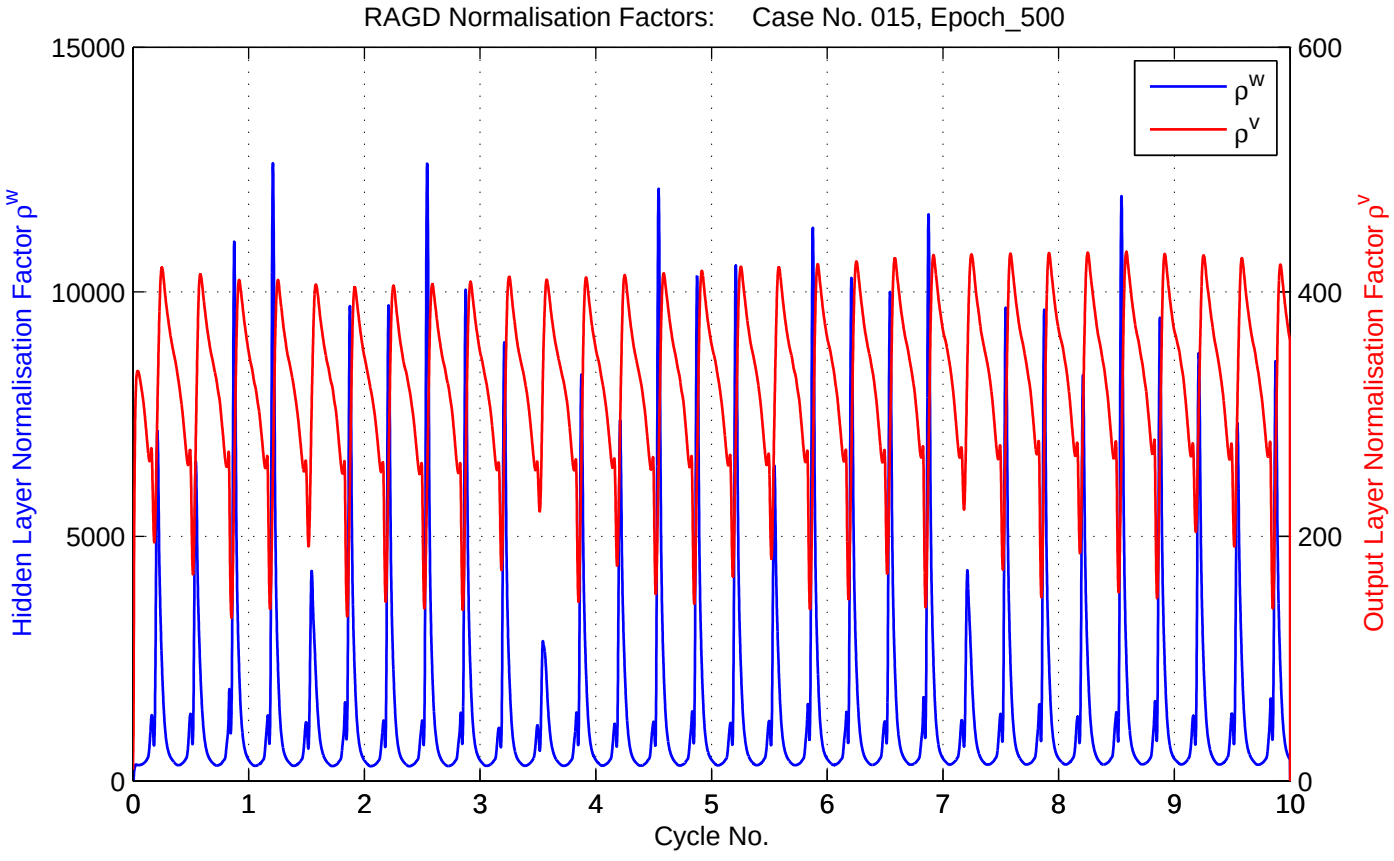

Unseen Cylinder Pressure: Case No. 015, Epoch_500
Target
Unseen Prediction
Normalised Cylinder Pressure
Cycle No.
1.25
1
0.75
0.5
0.25
0
−0.25
0
1
2
3
4
5
6
7
8
9
10

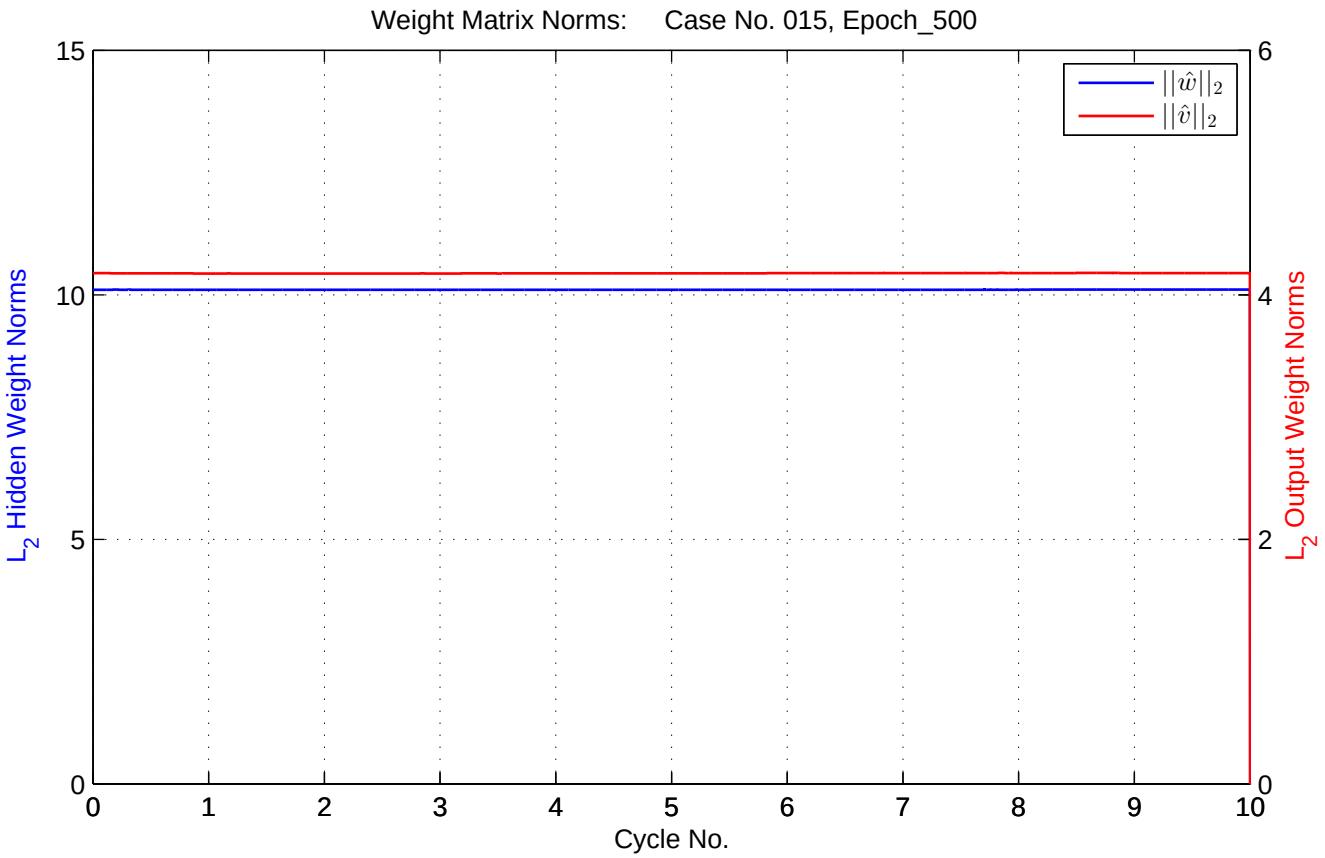

Percentage Error P_{max} Prediction: Case No. 015, Epoch_500
P_{max} Error [%]
Cycle No.
Training
Unseen

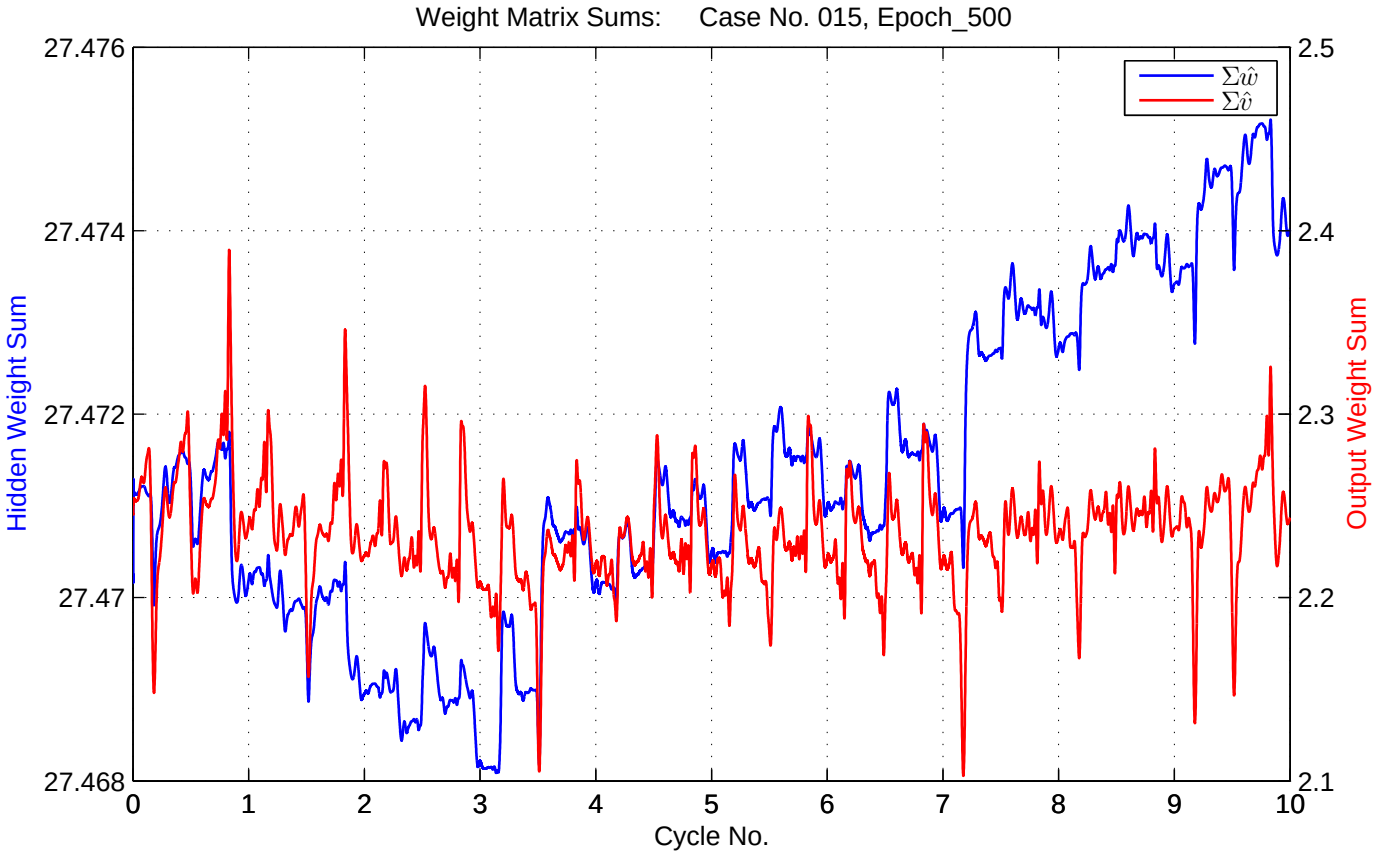

Error θ_{max} Prediction: Case No. 015, Epoch_500
θ_{max} Error [^{o}]
Cycle No.
Training
Unseen

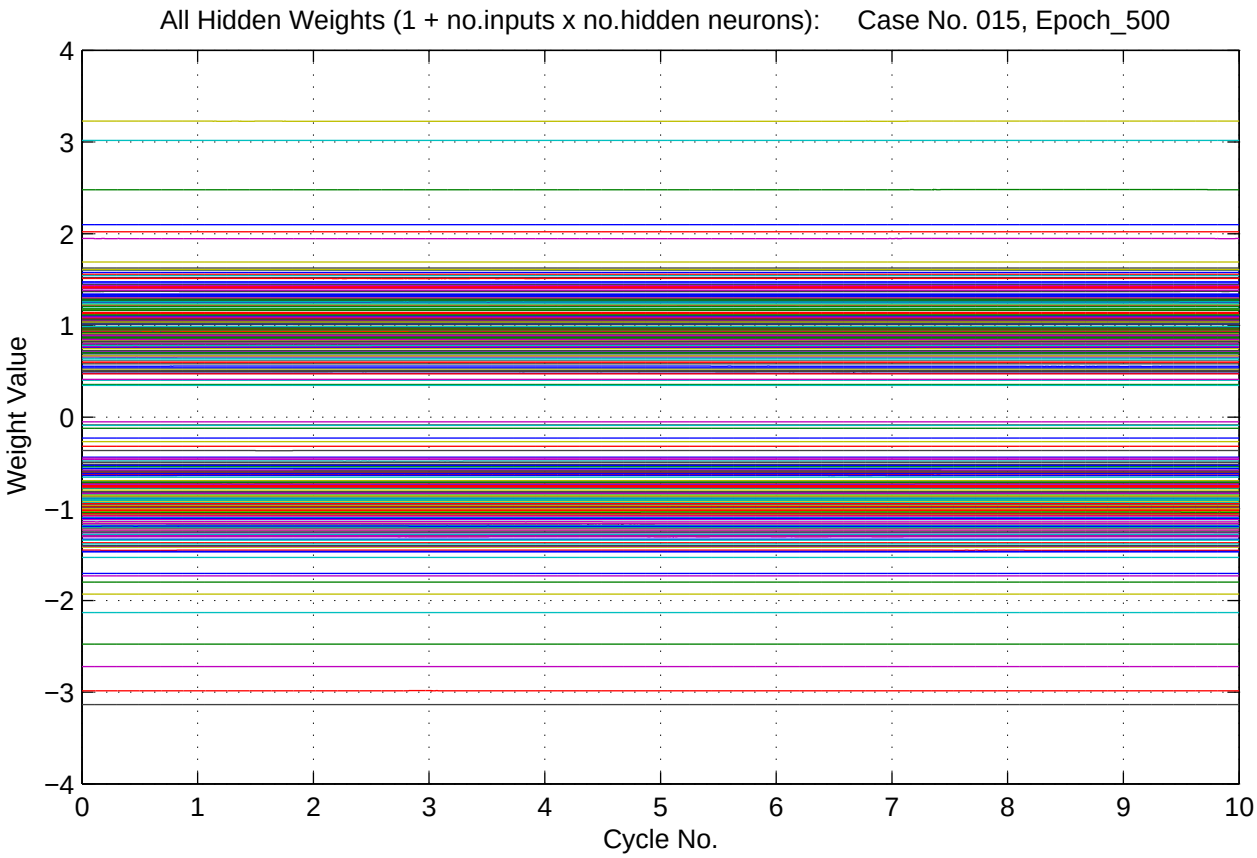

Prediction Error 10^oBTDC−110^oATDC: Case No. 015, Epoch_500
Normalised Pressure Error
Cycle No.
Training
Unseen

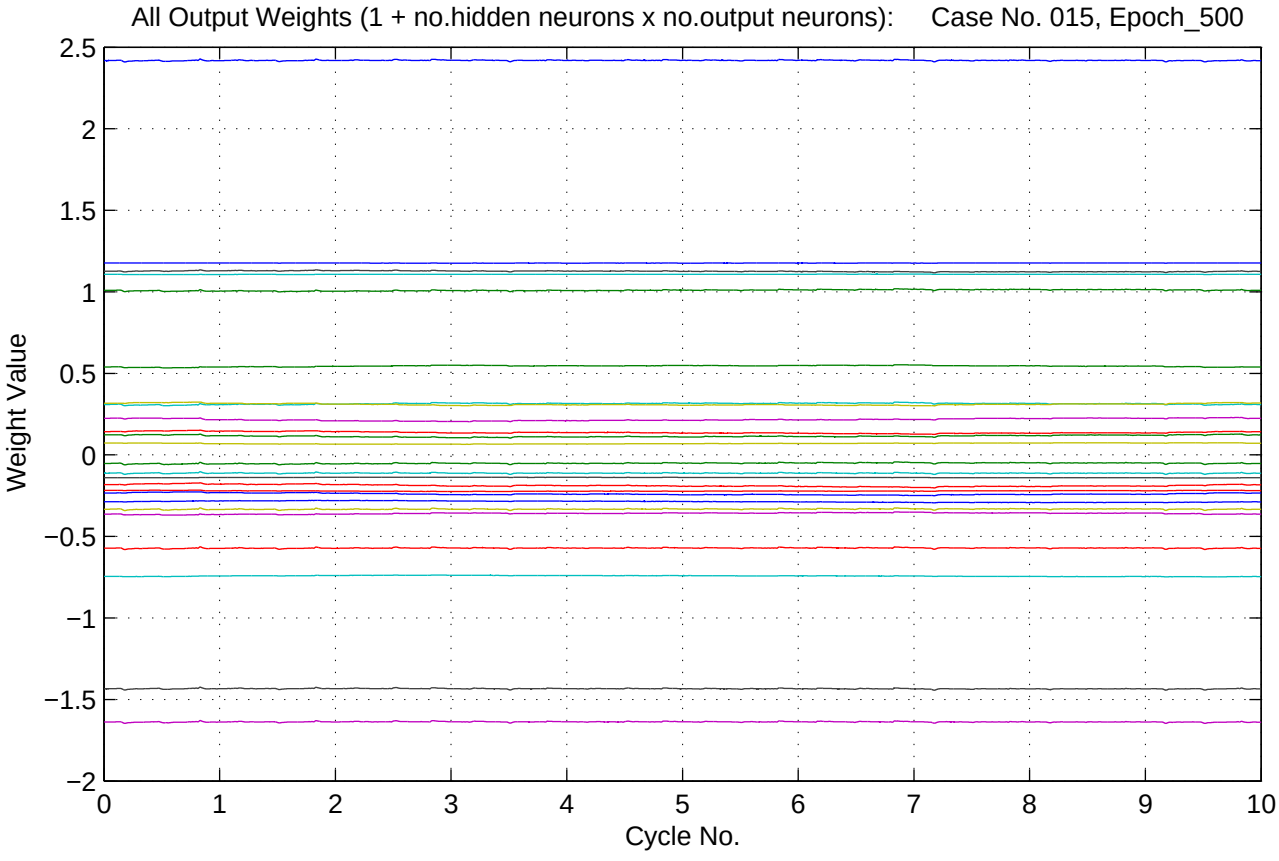

RAGD Switch Values: Case No. 015, Epoch_500
Switch Value
Cycle No.
1
0
1
0
1
0
1
0
0
1
2
3
4
5
6
7
8
9
10
α^w
α^v
β^w
β^v

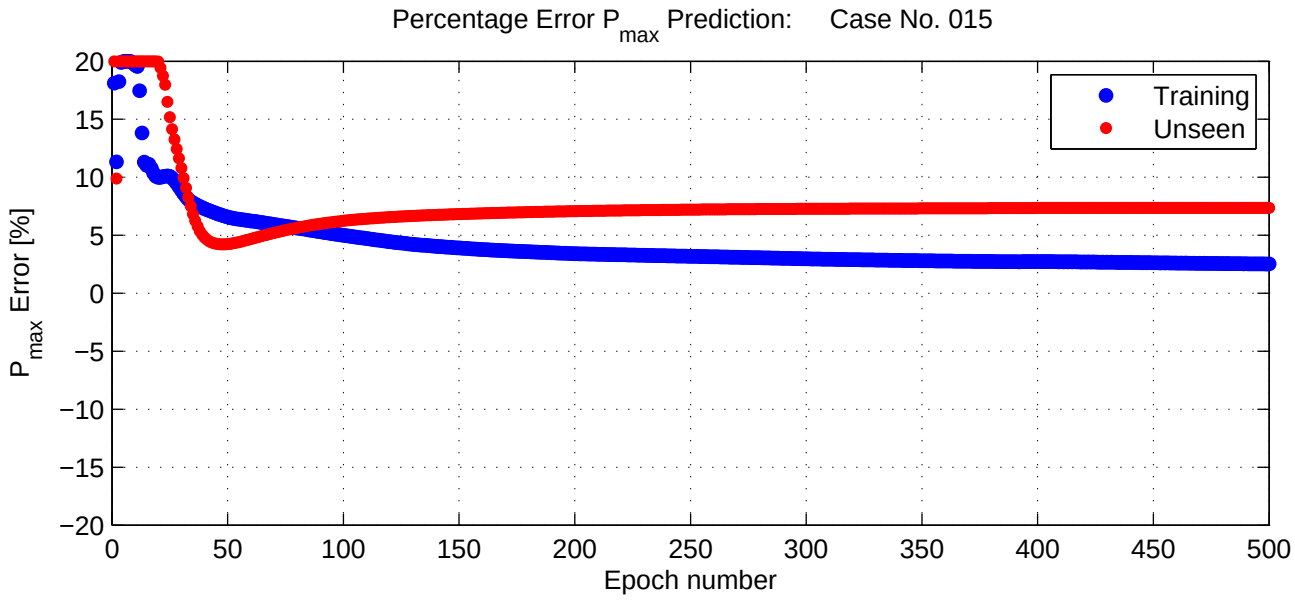

RAGD Normalisation Factors: Case No. 015, Epoch_500
Hidden Layer Normalisation Factor ρ^w
Output Layer Normalisation Factor ρ^v
Cycle No.
15000
10000
5000
0
600
400
200
0
0
1
2
3
4
5
6
7
8
9
10
ρ^w
ρ^v

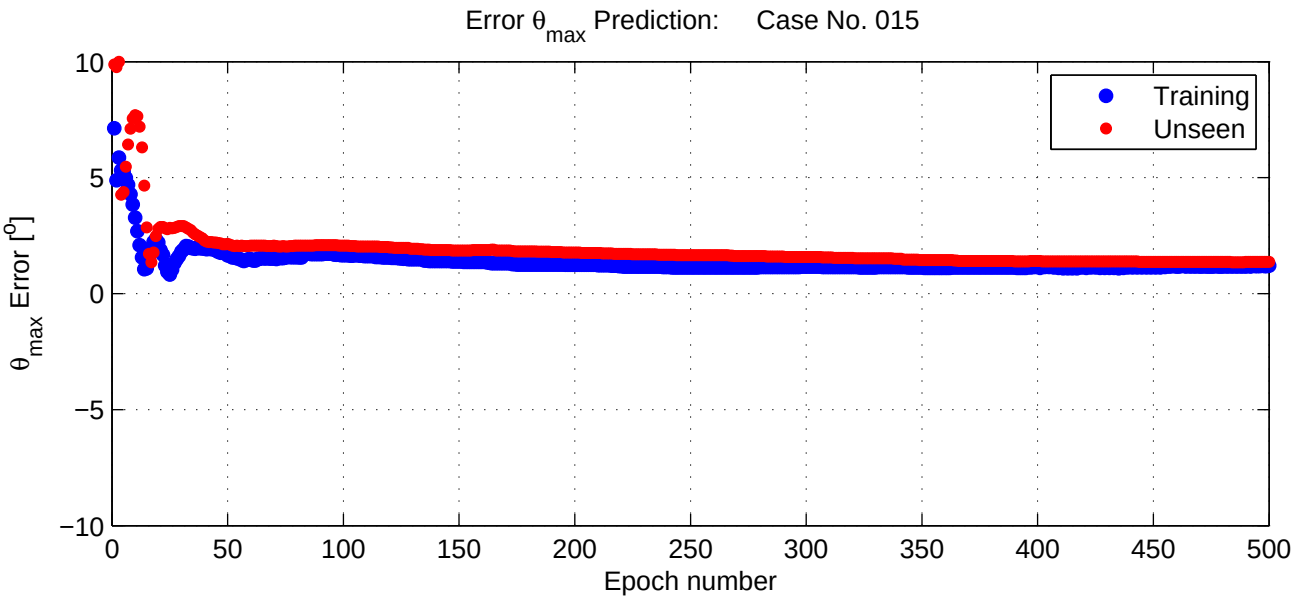

Weight Matrix Norms: Case No. 015, Epoch_500
$||\hat{w}||_2$
$||\hat{v}||_2$
L_2 Hidden Weight Norms
L_2 Output Weight Norms
Cycle No.
0
5
10
15
0
2
4
6
0
1
2
3
4
5
6
7
8
9
10

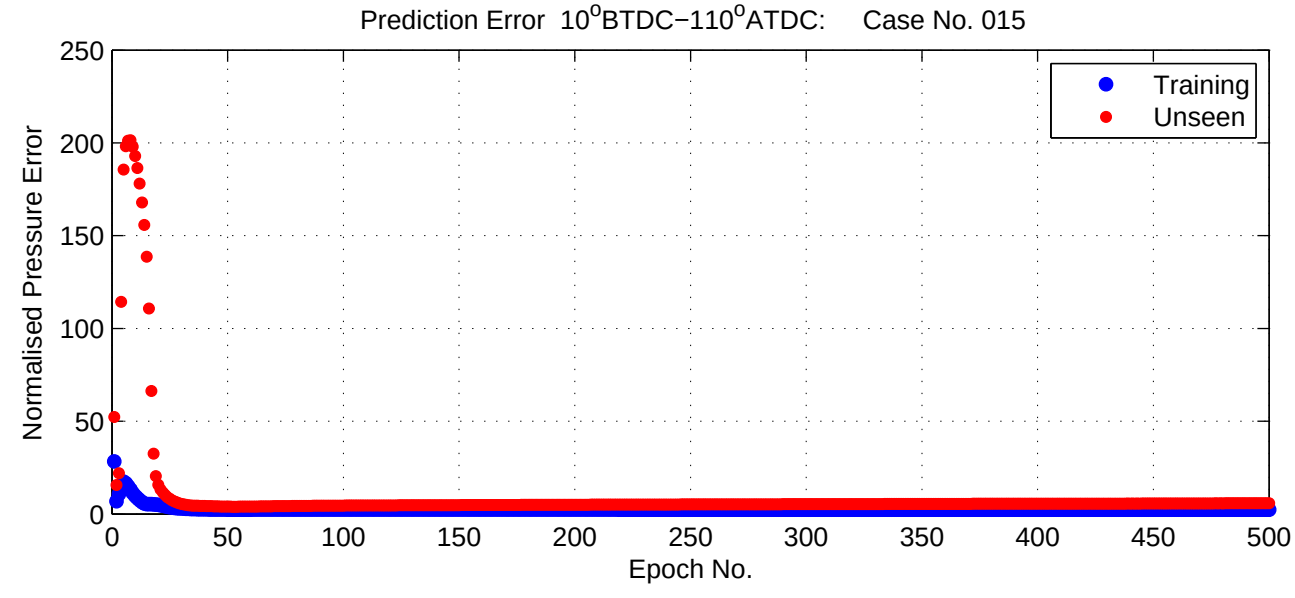

Weight Matrix Sums: Case No. 015, Epoch_500
$\Sigma\hat{w}$
$\Sigma\hat{v}$
Hidden Weight Sum
Output Weight Sum
Cycle No.
27.468
27.47
27.472
27.474
27.476
2.1
2.2
2.3
2.4
2.5
0
1
2
3
4
5
6
7
8
9
10

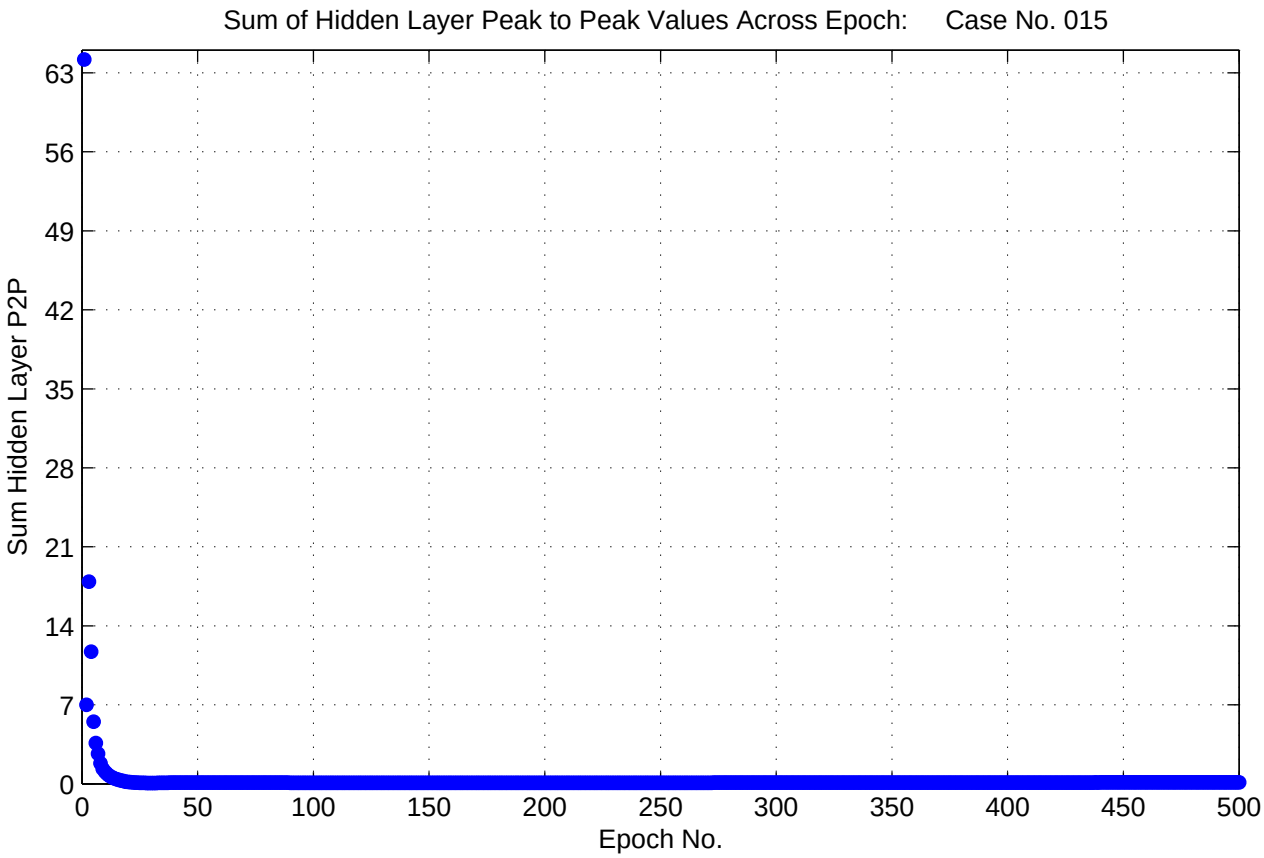

All Hidden Weights (1 + no.inputs x no.hidden neurons): Case No. 015, Epoch_500
Weight Value
Cycle No.
4
3
2
1
0
−1
−2
−3
−4
0
1
2
3
4
5
6
7
8
9
10

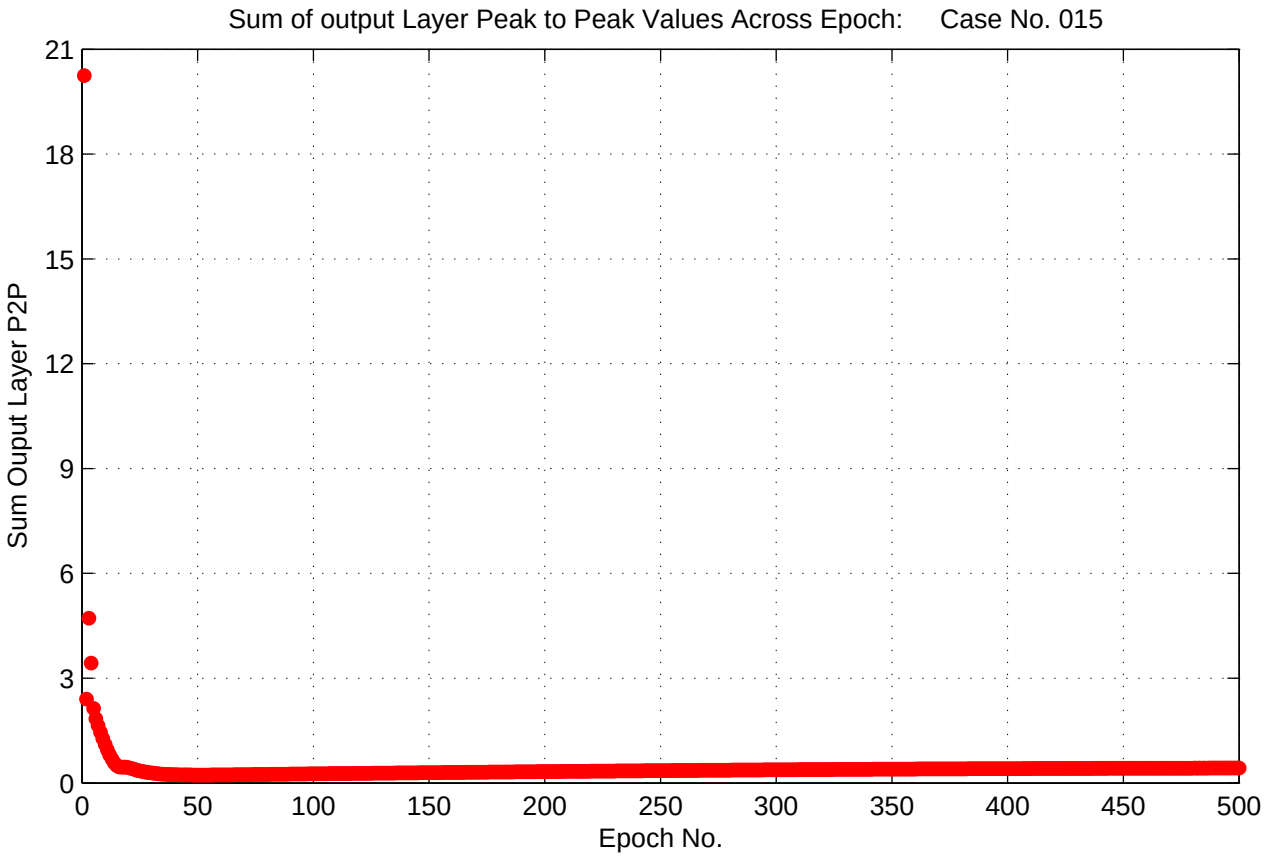

All Output Weights (1 + no.hidden neurons x no.output neurons): Case No. 015, Epoch_500
Weight Value
Cycle No.
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
1
2
3
4
5
6
7
8
9
10


Percentage Error P_{max} Prediction: Case No. 015
P_{max} Error [%]
Epoch number
Training
Unseen


Error θ_{max} Prediction: Case No. 015
θ_{max} Error [^{o}]
Epoch number
Training
Unseen


Prediction Error 10^{o}BTDC−110^{o}ATDC: Case No. 015
Normalised Pressure Error
Epoch No.
Training
Unseen


Sum of Hidden Layer Peak to Peak Values Across Epoch: Case No. 015
Sum Hidden Layer P2P
63
56
49
42
35
28
21
14
7
0
0
50
100
150
200
250
300
350
400
450
500
Epoch No.


Sum of output Layer Peak to Peak Values Across Epoch: Case No. 015
Sum Ouput Layer P2P
21
18
15
12
9
6
3
0
0
50
100
150
200
250
300
350
400
450
500
Epoch No.

Appendix B

Peak pressure metrics for NARX network trained on synthesised crank acceleration data with different numbers of hidden neurons.

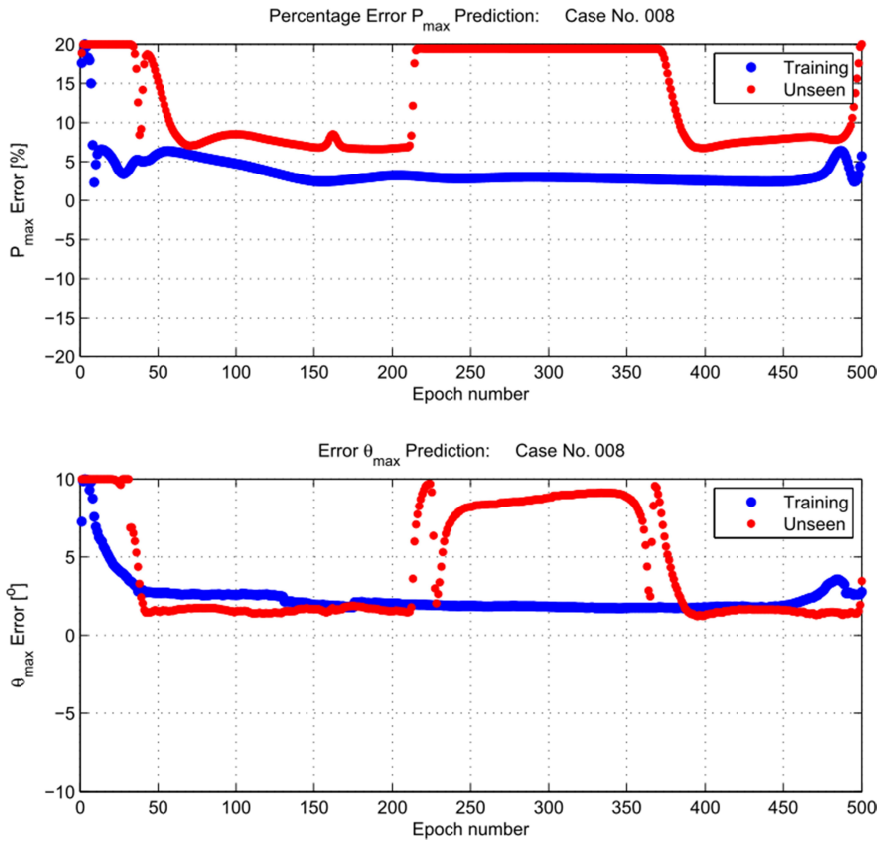


Figure B-1 – Mean peak pressure metrics over 500 epochs,synthesised crank acceleration inputs with 4 hidden neurons:1000 rev/min, 10 Nm

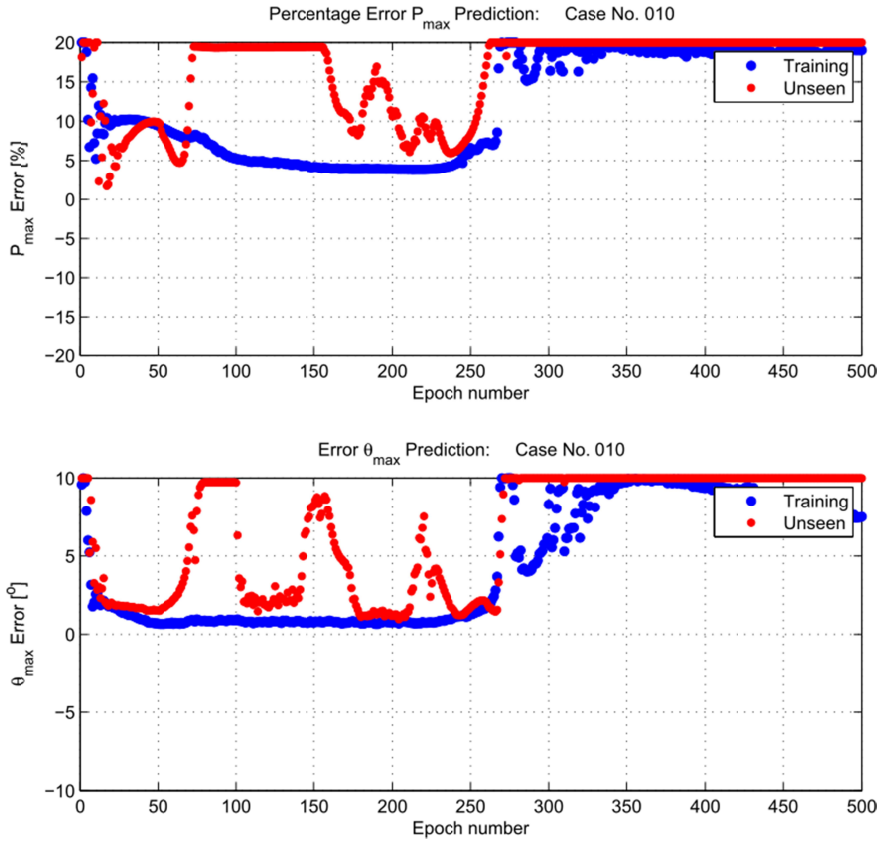


Figure B-2 – Mean peak pressure metrics over 500 epochs,synthesised crank acceleration inputs with 8 hidden neurons:1000 rev/min, 10 Nm

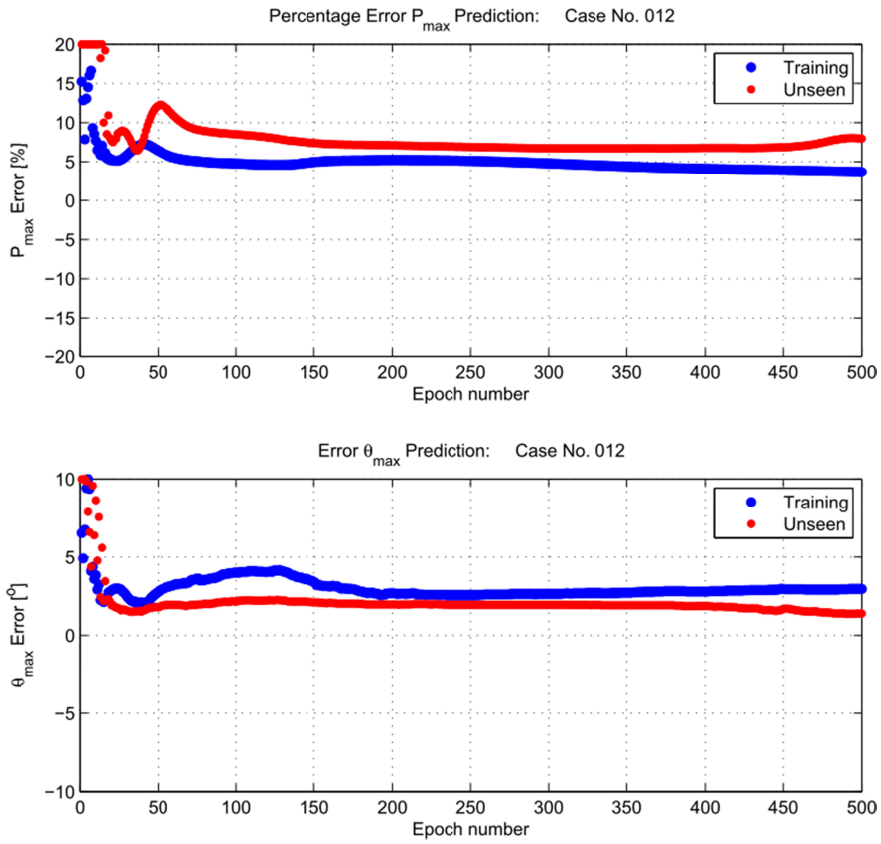


Figure B-3 – Mean peak pressure metrics over 500 epochs,synthesised crank acceleration inputs with 12 hidden neurons:1000 rev/min, 10 Nm

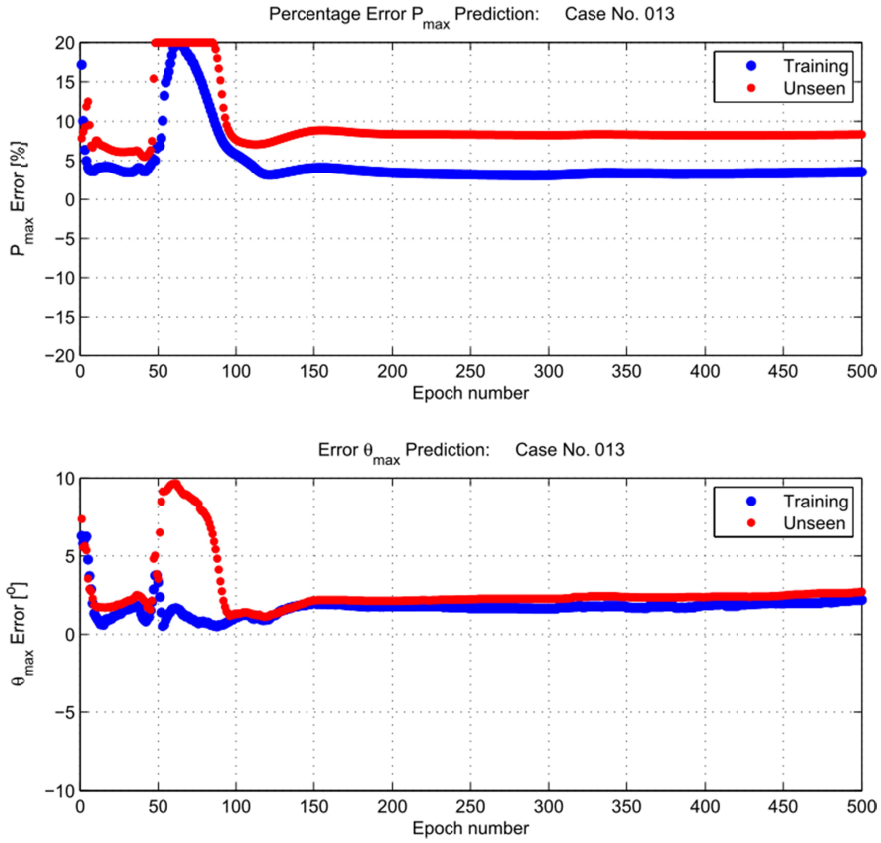


Figure B-4 – Mean peak pressure metrics over 500 epochs,synthesised crank acceleration inputs with 16 hidden neurons:1000 rev/min, 10 Nm

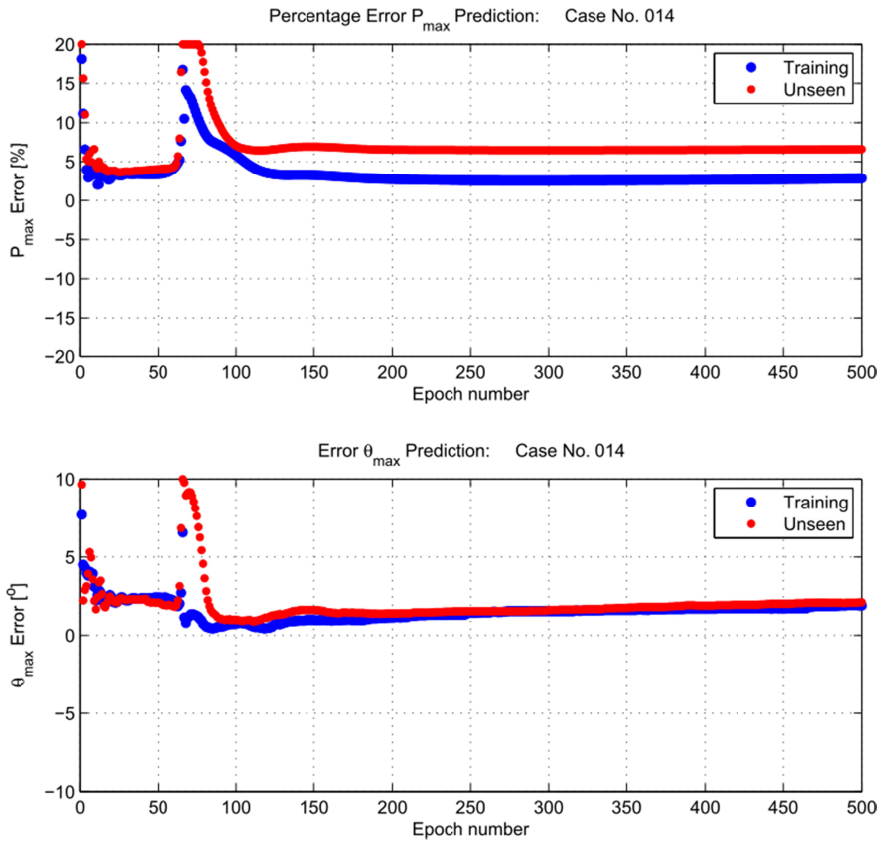


Figure B-5 – Mean peak pressure metrics over 500 epochs,synthesised crank acceleration inputs with 20 hidden neurons:1000 rev/min, 10 Nm

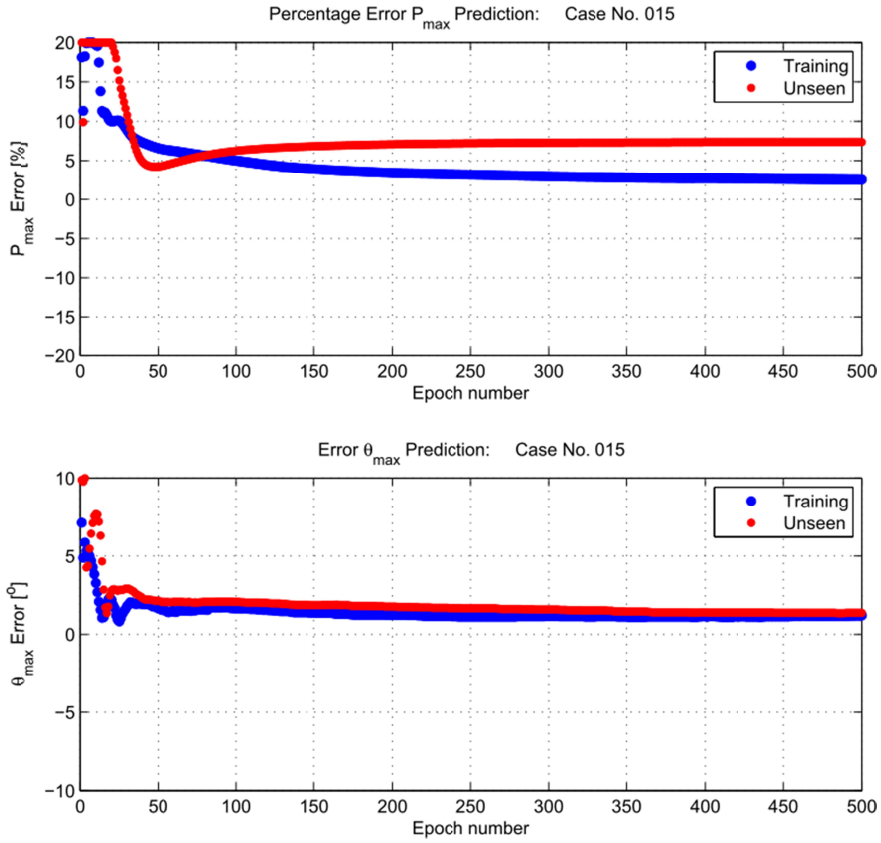


Figure B-6 – Mean peak pressure metrics over 500 epochs,synthesised crank acceleration inputs with 24 hidden neurons:1000 rev/min, 10 N